3-D SCENE RECONSTRUCTION FROM AERIAL IMAGERY

THESIS

Jared M. Ekholm, Captain, USAF

AFIT/APPLPHY/ENP/12-M03

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT/APPLPHY/ENP/12-M03

3-D SCENE RECONSTRUCTION FROM AERIAL IMAGERY

THESIS

Presented to the Faculty
Department of Engineering Physics
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Applied Physics

Jared M. Ekholm, BS, MA
Captain, USAF

March 2012

AFIT/APPLPHY/ENP/12-M03

3-D SCENE RECONSTRUCTION FROM AERIAL IMAGERY

Jared M. Ekholm, BS, MA
Captain, USAF

Approved:

	1 Mar 2012
Lt Col Karl Walli (Chairman)	Date
	1 Mar 2012
David Bunker, PhD (Member)	Date
	1 Mar 2012
Michael Talbert, PhD (Member)	Date

AFIT/APPLPHY/ENP/12-M03

Abstract

3-D scene reconstructions derived from Structure from Motion (SfM) and Multi-View Stereo (MVS) techniques were analyzed to determine the optimal reconnaissance flight characteristics suitable for target reconstruction. In support of this goal, a preliminary study of a simple 3-D geometric object facilitated the analysis of convergence angles and number of camera frames within a controlled environment. A series of 2-D images were acquired at convergence angles from 1° to 100° in 1° increments with the number of images varied from 2 to 20 at each angle. Reconstruction accuracy measurements revealed at least 3 camera frames and a 6° convergence angle were required to achieve results reminiscent of the original structure. Furthermore, improved results are realized with additional camera frames and expanded convergence angles enabling refinement of the focal length and camera motion estimation. The central investigative effort sought the applicability of certain airborne reconnaissance flight profiles to reconstructing ground targets. The data sets included images collected from a synthetic 3-D urban environment along circular, linear, and s-curve aerial flight profiles equipped with agile and non-agile sensors at look angles ranging from 0° (nadir) to 60° in 15° increments. S-curve profiles and dynamically controlled linear flight profiles produced the most diverse data sets resulting in superior reconstruction accuracy and density of points. Linear profiles equipped with non-agile cameras failed to reproduce identifiable results at near nadir look angles, but the results were dramatically increased when multiple orthogonal passes were combined and only overlapping images employed. Furthermore the effects of prominent images pivotal to the reconstruction processes were analyzed where a bimodal structure was observed relating the frequency of image use for each reconstructed 3-D vertex.

Table of Contents

List of Figures

Figure Page

Figure Page

Figure Page

List of Tables

Acknowledgements

First and foremost, I would like to express my appreciation to the faculty committee (Lt Col Karl Walli, Dr. David Bunker, and Dr. Michael Talbert) for their support and guidance throughout this effort. Thanks also go to the remainder of the team, including Dr. Ronald Tuttle, Dr. Christoph Borel-Donohue, Dr. Brian Tsou, and Todd Rovito whose encouragement and dedication were pivotal to the successful completion of this research. The computational effort would have never been successful without the daily support of Joseph Hendrix whose programming and computer skills are the envy of all. To each of you, your technical ingenuity, but more importantly your dedication to the pursuit of the unknown and attainment of knowledge will undoubtedly serve myself, future graduates, this institution, and our country beyond measurement.

Most importantly, to my beloved wife and boys who stood by my side through the highs and lows without fail. I am eternally grateful for your love, encouragement, and support over the past several years.

Finally to my father,
You instilled a love of science and discovery within me,
this work is dedicated to you.

Jared M. Ekholm

3-D SCENE RECONSTRUCTION FROM AERIAL IMAGERY

I. Introduction

Over the past decade overhead imagery has experienced exponential growth in both federal government and private sectors. While in its infancy, access to aerial imagery proved cost prohibitive to all but governmental organizations and multi-national corporations; however, recent advancements in space lift technology, commercial imagery, and the emergence of unmanned aerial vehicles resulted in a fundamental shift in the overhead imagery customer base. These radical evolutions significantly decreased both the cost and complexity of acquiring overhead or aerial imagery resulting in exponential demand growth as this imagery suddenly became available to the general public. Fueling this expansion has been the public's ravenous demand for situational awareness manifesting itself in multiple forms including professional, educational, and recreational uses. Professionals can quickly determine optimal sites for natural resource exploration and potential development sites. Educators can explore natural landmarks and experience foreign countries with their students, while recreational users can be immediately transported to the remotest areas of the world through software tools such as Google Earth [6].

The public's demand for this information has fueled several research areas focusing on extracting additional information from existing data sets. Two paramount research threads, Structure from Motion (SfM) and Multi-View Stereo (MVS), involve deriving 3-D structure information from 2-D images. Past researchers have developed the necessary photogrammetric techniques to extract this information from both electro-optical and multi-modal imagery [28, 23, 35] and now applications for these techniques are being found in numerous scientific ventures including synthetic scene regeneration and city and terrain modeling. The implications of this research are far-reaching especially when used to augment remote sensing efforts.

1.1 Motivation

Traditionally depth information is derived from either direct ground measurements or more commonly, active detection and ranging efforts such as LIDAR or RADAR which measure the ground reflections from emitted electromagnetic pulses. These methods provide an extremely accurate mapping of the scene and have been the gold standard by which 3-D scene reconstructions are measured, but their use is not applicable or available in many situations. For instance, the collector is required to actively interrogate the scene and requires advanced transmitters and receivers to perform the collection. This limits the availability by which one can obtain the required depth information as these systems are rare and existing high fidelity global coverage is very limited. Additionally, areas denied to manned or unmanned air vehicles preclude the 3-D mapping of the scene via traditional techniques. However, overhead space assets collecting 2-D electro-optical imagery are not limited in this regard, and thus the ability to extract 3-D information from 2-D electro-optical imagery is of immense important to the military, intelligence community, and national decision makers. Therefore it is imperative to develop the methodology to derive 3-D depth information using the plethora of existing electro-optical sensors.

SfM's immediate consequence to both the military and the private sectors is the enablement of computer vision, thereby opening the realm of autonomous navigation in cluttered environments. In 1995, Dickmann demonstrated the autonomous navigation of a passenger vehicle at speeds of 110 mph while simultaneously maintaining lane control, reading traffic signs, and passing slower traffic [28]. Beyond robotic vision, SfM possesses the potential to mitigate problems found in a number of disciplines specific to military operations and intelligence. With increased reliance on precision targeting, our nation's warfighting capability heavily relies on remotely sensed intelligence products from airborne and satellite platforms. The fusion of these disparate data sets allows for in-depth analysis of sensitive targets. For example, fusion of SfM and hyperspectral imaging (HSI) culminates in revolutionary advancements in remote sensing. Independently, HSI collects reflected and emitted spectra from target objects facilitating target detection and object classification, camoflauge defeat, vegetation analysis, and disaster assessment. Despite HSI's extraordinary abilities, problems such as shadowing, variable illumination loading, occlusions, and parallax effects hinder target recognition and extraction of scene information. These problems lead to inaccurate hyperspectral measurements and possible misidentification of target materials. Previous efforts have validated the fusion of HSI with LIDAR datasets to mitigate the 3-D influences [35]; however, LIDAR collection systems rarely accompany hyperspectral collection systems and existing datasets are relatively nonexistent. Fortunately, SfM presents a viable alternative to LIDAR.

SfM development benefits the commercial and industrial sectors as well. Improvement of hyperspectral registration and classification allow damage response coordinators to quickly determine the extent of structural damage on a city wide scale. Archaeologist's reconstruction of ancient sites can forgo an intensive process of erecting a grid and documenting the placement of all finds, and medical surgeons can gauge depth while performing laparoscopic or endoscopic surgeries [33]. In all, it is difficult to determine a field not impacted by the tremendous promise of 3-D scene reconstruction.

As an demonstration of SfM's ability to reconstruct complex scenes, the Air Force Institute of Technology's central campus was reconstructed using images acquired in an unstructured pattern using a common consumer grade camera. Areas void of vertices represent either non-imaged scenes or areas obscured by trees or other vegetation.

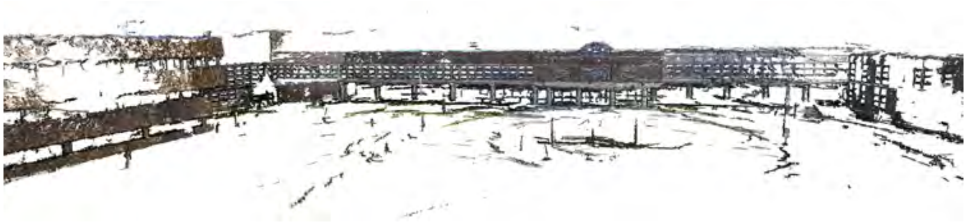


Figure 1. 3-D reconstruction of AFIT's main campus. The reconstruction contains 3,979,668 3-D vertices generated from 241 images.

1.2 Problem Statement

The collection of source imagery required to reconstruct a 3-D scene has included controlled laboratory collection, limited overhead collection efforts, and massive Internet wide sourcing resulting in hundreds of thousands of images [20]. These efforts validate the reconstruction algorithm's ability to reconstruct the imaged scene; however, little if any analysis exists to determine the optimal airborne collection characteristics necessary for an accurate reconstruction. Tradeoffs between loiter time, proximity to target, number of images, angular diversity, and convergence angle on the reconstructed scene's approximation to the real world environment have yet to be determined. If 3-D reconstruction techniques are to be applied to everyday scene reconstruction and remote sensing problems these relationships must be quantified.

1.3 Research Objectives

This thesis presents a fundamental analysis of the image collection requirements and reconstruction of accurate 3-D scene models derived from 2-D imagery sets.

Specifically the relationship between the final 3-D model and various collection parameters will be investigated.

While the objectives of this research are threefold, it focuses primarily on the accuracy of the reconstructed scene model compared to actual scene dimensions and geometry. The salient principle governing the final reconstruction is the quality of input data. The linear and non-linear methods employed by the SfM and MVS algorithms require comprehensive and diverse data sets to achieve optimal solutions. Therefore the three specific research areas to be investigated include: effects of convergence angles, number of camera frames, and finally the effects of variable viewing angles stemming from typical airborne reconnaissance flight profiles to include angular diversity, static versus dynamic imaging, and target visibility. To investigate these three characteristics, a multi-phase effort is required in which the initial phase explores the fundamental principles inherent to the reconstruction. The second phase analyzes the effects of variable viewing geometries to include angular diversity and magnification of the target within the data sets. As mentioned earlier the reconstruction methods are highly dependent on a diverse input data. Data sets containing variations in the both the viewing geometry and spatial dimensions of the target structure should provide the highest quality results. The necessary aerial reconstructions required to obtain this degree of diversity within the dataset will be the salient feature of this effort.

II. Theory

2.1 Chapter Overview

This chapter presents the fundamental techniques required for reconstruction of a 3-D scene. This process, commonly referred to as Structure from Motion (SfM) within the photogrammetry and computer vision research areas, has experienced revitalized attention in recent years. Research threads such as autonomous navigation, situational awareness, and terrain mapping have incorporated themselves into everyday commercial products such as robots, proximity sensors in vehicles, and advanced image processing. Such efforts have provided the means for robots to navigate hallways, vehicles to autonomously navigate along interstate routes at speeds in excess of 110 mph, and the inclusion of the first down line in televised football games [28].

These technological advancements are made possible through 3-D vision and the exploitation of epipolar geometry to extract depth geometry from multi-view images. Within the framework of presenting the steps required for 3-D reconstruction, contributions from past and current researches will be incorporated in conjunction with mathematical tools and algorithms used to recreate the 3-D images seen in the following chapters.

This chapter consists of two complementary components. The first half is limited to a qualitative overview of visual perception and epipolar geometry. It is hoped by limiting the discussion to the general topics the reader will gain the necessary overview to fully understand the following mathematical development in the latter half. The second half presents an incremental process to recover depth information with corresponding algorithms to assist.

2.2 Visual Perception: The Human Eye

The human eye defines our understanding of depth perception and serves as the foundation for the implementation of 3-D computer vision. The combination of iris, lens, and retina are analogous to a mechanical imaging system consisting of an aperture, optical lens, and imaging focal plane. Therefore a brief explanation to the underlying mechanisms responsible for human depth perception aid immeasurably to the final understanding of machine vision.

2.2.1 Accommodation and Convergence.

The eye's ability to perceive depth stems from two functions, accommodation and convergence [32]. Accommodation results from the eye's ability to temporarily distort the lens via the ocular muscles to focus a real world object on the retina. Depth information is extracted from different focal lengths required to focus scene objects and provides absolute, or qualitative, depth information. Typically, accommodation is regarded as a weak source of depth perception and only applicable at close ranges within eight feet of the individual. The ability to derive depth information based on the focal length alone explains why marginal depth perception is retained with a single eye.

Convergence provides a much more powerful means of depth perception and is measured as the angle the eyes turn toward one another to focus on nearby objects. Distant objects require little convergence, conversely, nearby objects require significantly more deflection to fixate on the object. Similar to accommodation, convergence provides absolute depth information with respect to viewer and is limited to ranges less than six to eight feet [32]. The range limitations are due to the negligible effect of distance on convergence angle as seen in Figure 2.

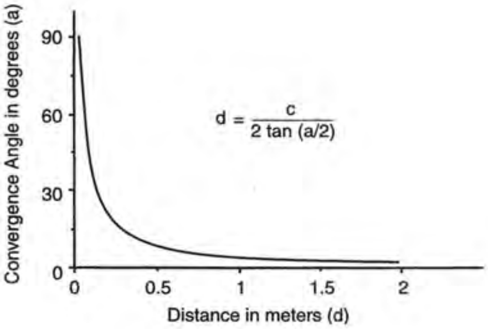


Figure 2. With a fixed human pupillary baseline, *c*, of 2.5 inches, convergence angles, *a*, sharply decrease at near distances, but exhibit little variation at distances, *d*, beyond 8 feet or approximately 2.5 meters [32].

Furthermore, Figure 3 provides a visual depiction of this phenomena where the convergence angle is defined as the angle between two rays emanating from the object and passing through the left and right eye. θ_1 and θ_2 represent two such angles. In humans the pupillary distance (PD) is fixed. However in remote imaging situations, the distance between the camera centers is variable, and the convergence angle becomes an important parameter in optimizing scene reconstruction accuracy.

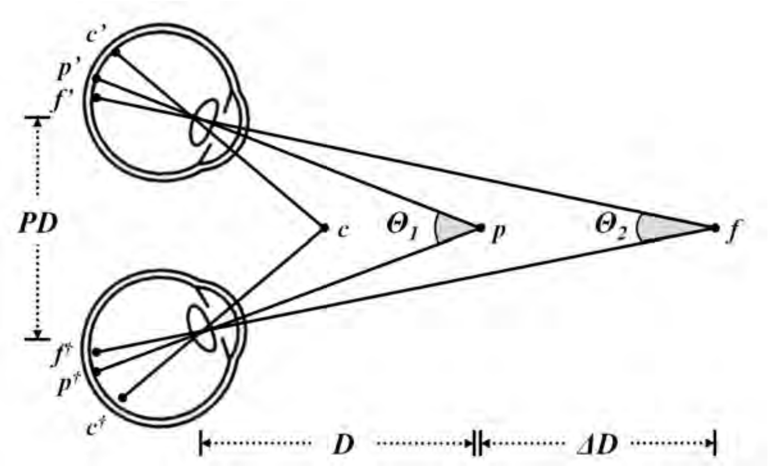


Figure 3. Human vision relating both convergence and stereopsis. The convergence angles, θ_1 and θ_2, change as a function of D. Furthermore the lateral displacement on the retinas depending on whether nearby objects are closer (c) or further (f) than the focusing point (p) resulting in a stereopsis effect. Image adapted from Palmer [32].

Both accommodation and convergence provide accurate and absolute depth information to facilitate interactions with nearby objects. However, depth perception at longer distances is subjugated to the area of stereopsis.

2.2.2 Stereopsis and Parallax Effects.

Stereopsis relates the lateral displacement, or binocular disparity, between similar images on the retinas to perceive depth. Although similar to convergence, stereopsis is distinctly different in that it only provides a relative distance between objects. Since objects surrounding the focused object appear at different locations on the retina, the magnitude and direction of the lateral displacement invoke a sense of relative depth perception. This phenomena can be seen by referencing Figure 3 and noting the position of the three points, c, p, and f. The eyes fixate on point p and the corresponding images are projected onto the center of the retina as p' and $p^\dagger$. Similarly, the close (c) and far (f) points project on the retina, but in different relation to projections p' and $p^\dagger$. The magnitude and direction of the lateral displacement between $p' \to c'$ and $p^\dagger \to c^\dagger$ allows for relative depth perception.

Stereopsis stems from the parallax effect which is the relative displacement of an apparent object when viewed along two different sight lines. This effect can be directly experienced by extending one's finger at arm distance and aligning it with a distant point. By slightly shifting one's head, the apparent position of the finger seems to move in relation to the distant point although both remain stationary. Furthermore by viewing the same setup with both eyes open and focusing on the far wall, two fingers appear slightly offset from one another. This offset represents the lateral displacement of the object on the each retina.

As a mathematical example, the minimum separation distance required to perceive two objects at differing depths can be determined using a similar scenario as that as seen in Figure 3. The human pupillary distance (PD) averages 2.5 inches while the typical difference between angles which can be resolved by the retina is 0.008° [29]. Mathematically, McNeil showed the minimum depth which can be stereoscopically achieved, ΔD, is calculated by

$$\Delta D = \frac{D^2 \times \tan(\eta)}{PD - D\tan(\eta)} \approx \frac{D^2\eta}{PD - D\eta} \tag{1}$$

where PD is the pupillary distance, D is the distance to first object, and η is the binocular disparity as defined as $\theta_2 - \theta_1$. Note the tangent small angle approximation, $\tan(\theta) \approx \theta$. For example, when two objects are displaced 10 feet from the viewer, the objects must be separated by 0.83 inches to perceive a difference in depth between the objects.

The stereoscopic arrangement in Figure 3 infers a mechanical setup with two distinct cameras or a single translating camera capturing multiple frames along its path. Focal plane arrays image real world objects which are recorded as pixels (p and p') on each image. The relationship between the correspondences, $p' \mapsto p^\dagger$, $c' \mapsto c^\dagger$, and $f' \mapsto f^\dagger$ forms the basis for extracting depth information from 2-D images. Clearly, stereopsis serves as the stepping stone by which human perception relates to computer vision.

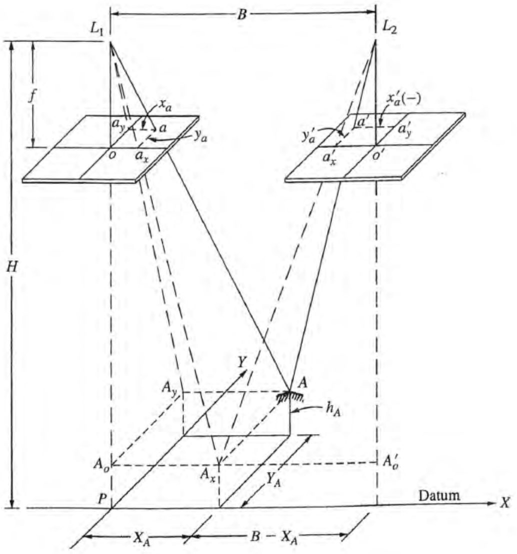


Figure 4. Stereopsis applied to computer vision. Using two overlapping frames acquired at equivalent vertical heights, structure depth can be extracted through simple ray tracing and trigonometry [36].

In Figure 4 two frames, analogous to the human retina, acquire overlapping images of a structure with a height h_A above the datum from equivalent distances. Using point P as the origin, Wolf and Dewitt showed the structure height is dependent on the baseline distance (B), distance from camera to datum (H), camera focal length (f), and the positions of the point on each of the focal arrays as seen in the following formula [36].

$$h_A = H - \frac{Bf}{x_a - x'_a} \tag{2}$$

Furthermore, the equivalent ground distances may also be extracted through the following equations.

$$X_A = B\frac{x_a}{x_a - x'_a} \tag{3}$$

$$Y_A = B\frac{y_a}{x_a - x'_a} \tag{4}$$

These three equations are commonly referred to as the parallax equations and serve as the preliminary principles to derive 3-D structure. However, the equations are limited in that the camera motion must be linear, maintained at constant height above the datum, and the focal plane parallel to the datum. These requirements introduce significant limitations which may prevent collection in operationally sensitive situations. Fortunately, epipolar geometry offers a generalized approach to achieve the same results without these limiting conditions.

2.3 Epipolar Geometry Introduction

The ability to extract an additional dimension from 2-D images is possible through epipolar geometry. Epipolar geometry, or the geometry of stereo vision, describes the relationship between two images of the same real world scene. In principle given two distinct images of the same 3-D scene, geometric relationships between the 2-D image points allow the derivation of a mathematical relationship between the image points. At the most basic level this relationship is defined within the homography matrix, H_π, which captures the rotation and translation between images. Visually this is represented in Figure 5.

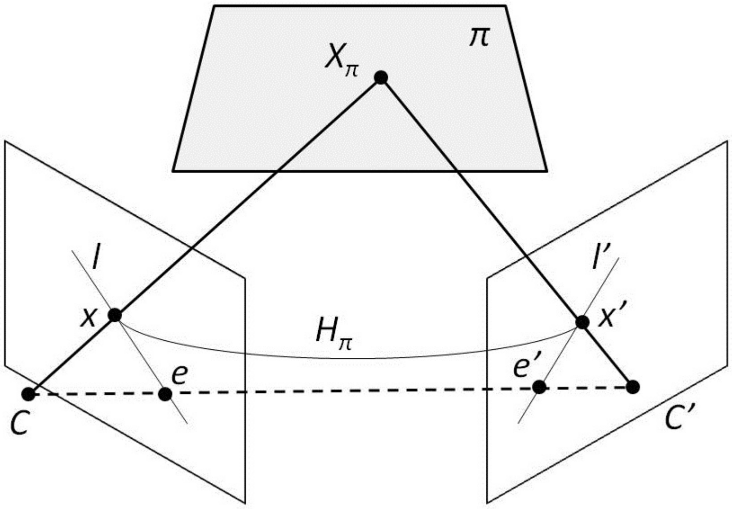


Figure 5. Basis of epipolar geometry [23].

Within Figure 5, two cameras image a real world scene, represented by plane π, in which a unique feature X_π such a corner of a window or tip of a steeple, is imaged on each camera's focal plane array as x and x'. For simplicity the two cameras will be referred to as the left and right camera and only the left camera analyzed, although the same logic applies for the camera on the right. Since the camera centers, C and C', are distinct, they are mapped to the imaging plane of the other camera. These locations are termed epipolar points and denoted as e and e'. At this point it is important to note the imaging plane exists as a theoretical plane extending beyond the metric dimensions of the camera's focal plane array. Therefore e and e' may not be located on the actual sensor unit of the camera. At this point an epipolar line, l, connecting e and x represents the projection of the ray joining X_π to x' and C'. From the left camera, the ray between C and X_π is seen as a point; however, in the right camera this same ray maps to the epipolar line, l'. Symmetrically the inverse is true where the ray from X_π to C' is viewed as l on the left camera. Therefore, the epipolar line is a function of the 3-D point X_π and all epipolar lines in one image must intersect the epipolar point of that image. In effect when relating two images the epipolar point is the source of radiating epipolar lines relating correspondences from image to image. Figure 6 depicts this relationship.

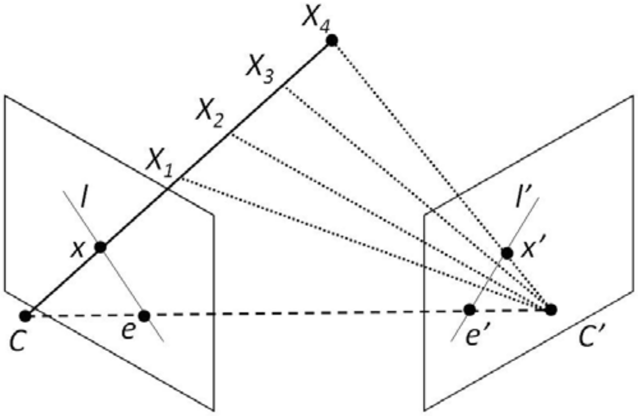


Figure 6. Projection of epipolar lines [23].

If the projection points, x and x', and camera centers, C and C', are known then the epipolar line l' can be determined. Since point X_n projects onto the left image at point x the correspond point, x' must lie on the epipolar line l' on the right image. Depth information from X_n can be inferred by triangulating the intersection of $\overrightarrow{Cx}$ and $\overrightarrow{C'x'}$. In summary, extraction of the corresponding image points and camera centers is the pivotal data required for epipolar geometry and 3-D reconstructions, and the pursuit of their determination is a main focus point.

In reality, the camera pose is often unknown and aberrations from the camera's imaging optics, such as pincushion and barrel distortion, alter the perceived epipolar lines and frustrate direct inference of epipolar lines from corresponding image features. Therefore image distortion correction becomes the first step in any reconstruction effort; however, methods of correcting distorting images will not be addressed in this effort. Without a priori knowledge of of the cameras, the images must be related to one another by in-scene information alone upon which the fundamental matrix, F, which relates the two images is developed.

For the purposes of this effort we will assume camera parameters are unknown and fundamental matrix reconstruction techniques will be employed. Hartley and

Zisserman's text, *Multiple View Geometry*, provides the necessary steps to derive an initial guess of the 3-D point location [23]. The process consists of three steps to estimate a 3-D point location from the correspondences, $x \mapsto x'$, between each image.

- Derive the fundamental matrix, F
- Derive the camera matrices, Π_1 and Π_2, from F
- Estimate the 3-D coordinates, X, from Π_1 and Π_2 as well as the 2-D points, x and x'

With these steps it is possible to obtain an initial estimate for the sparse 3-D structure of the object. In order to further refine the 3-D estimates, non-linear regression analysis is performed using all possible correspondences in a step termed bundle adjustment in which all 3-D vertices and camera centers are optimized to minimize reconstruction error. The Levenberg-Marquart technique provides a robust method for minimizing the reconstruction problem, and ultimately offers an accurate sparse reconstruction consisting of hundreds of 3-D points derived from a series of images measuring 640×400 pixels. Further analysis can result in a dense reconstruction by searching for additional correspondences along the epipolar lines with knowledge of the C and C' resulting in an order of magnitude increase in the number of 3-D points.

2.4 Recovering 3-D Depth Information

The preceding section frames the analytical discussion below. In this section the reconstruction process is broken down into distinct components beginning with feature extraction to the final non-linear regression techniques to estimate depth information.

2.4.1 Approach.

The basic premise to recovering 3-D scene information from multiple view geometry consists of five steps:

- Image generation
- Feature extraction from imagery
- Derive correspondences between extracted image features
- Derive initial group relationship and perform sparse bundle adjustment
- Incorporation of derived camera parameters for dense point cloud reconstruction

Within this process variants exist depending on foreknowledge of scene geometry and camera parameters. For instance, inclusion of the camera's IOPs and EOPs (internal/external operating parameters such as the camera calibration parameters (K) and real world camera pose, $[R, T]$, within the SfM and MVS pipelines provide real world knowledge and allow for absolute or real world reconstruction. Note the parameters, K, R, and T are discussed in detail in the following section. However for the purposes of this research, a priori knowledge of these parameters will not be included to generalize the reconstruction problem thereby allowing future readers the ability to obtain 3-D reconstruction independent of prior camera or scene knowledge.

2.4.2 Image Generation.

Under the assumptions of a pinhole camera approximation and lambertian surfaces, the issue of image formation is reduced to simple ray projection where only three transformations must be accounted for: transformation between camera and world frames, projection of 3-D world coordinates to 2-D image coordinates, and transformation between image coordinate frames [28]. Under realistic conditions aberrations introduced by optical elements, image noise, and peculiarities in the actual camera focal plane introduce second and third order effects which may confuse the relationship between the real world coordinate and recorded image point; however, under this discussion an ideal perspective camera is considered.

The first transformation between camera and world frames is governed by the rigid body transformation $g = [R, T]$ of the real world coordinate, X_o,

$$\mathbf{X} = R\mathbf{X_o} + T \tag{5}$$

where X is the same relative point X_o with reference to the camera frame and R and T are the required 3×3 rotation matrix and 3×1 translation vector between the camera and world frames. The rotation and translation components are frequently combined into homogeneous coordinates as seen below.

$$g = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \tag{6}$$

The second transformation is the projection of the 3-D world coordinates onto the 2-D image plane. The point X is projected onto the image plane through the relationship,

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{7}$$

where f is the focal length of the camera system, and when expressed in homogeneous coordinates Equation 7 can be written as

$$Z\begin{bmatrix}x\\y\\1\end{bmatrix}=\lambda\begin{bmatrix}x\\y\\1\end{bmatrix}=\begin{bmatrix}f&0&0&0\\0&f&0&0\\0&0&1&0\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}\in\Re^{4\times4} \tag{8}$$

Note in Equation 8, the Z coordinate remains unknown and therefore is assigned as the positive scalar, λ.

The third and final transformation involves the intrinsic camera parameters where the ideal image coordinates $\mathbf{x}=[x,y,1]^T$ are related to the actual image coordinates $\mathbf{x}'=[x',y',1]^T$. Intrinsic camera parameters such as pixel dimensions may also be included by modifying the above matrices to define the camera calibration matrix,

$$K=\begin{bmatrix}fs_x&fs_\theta&o_x\\0&fs_y&o_y\\0&0&1\end{bmatrix} \tag{9}$$

where s_x and s_y are the dimension of image in pixels, s_θ skew of the pixel (commonly zero), and o_x and o_y represent the optical center of the image plane which is typically the center of the recorded digital image. This matrix is pivotal in the subsequent steps to include final Euclidean reconstruction and many of the reconstruction algorithms are greatly simplified when foreknowledge of the camera's IOPs are known and inserted into K. However knowledge of K is not required as methods can suggest or infer the matrix when minimizing the final reconstruction. An excellent resource for determining this matrix is the MATLAB camera calibration toolbox created by Jean-Yves Bouguet [3]. Furthermore, autocalibration techniques may be employed to determine the camera calibration matrix from scene information alone and substituted at this point to obtain partially calibrated cameras. These techniques utilize the absolute quadric constraint and require an initial focal length estimate. Autocalibration techniques will be briefly addressed in this effort, but Yi Ma provides an excellent discussion in his text [28].

Finally when all three transformations are included the overall image formation is governed by

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_o \\ Y_o \\ Z_o \\ 1 \end{bmatrix} \tag{10}$$

In shorthand Equation 10 can be written as

$$\lambda \mathbf{x}' = K\Pi_o \mathbf{X} = K\Pi_o g \mathbf{X_o} \tag{11}$$

where Π_o is a 3×4 identity matrix seen in Equation 10.

The parameters, K, R, and T, are the core variables determined within the SfM process. In a controlled collection environment prior knowledge of the camera calibration matrix, K, exists or can be easily ascertained using existing camera calibration methods mentioned earlier. Additionally, the rotation and translation components can be determined through GPS coordinates and internal navigation systems which relay the exact coordinates and projection vectors between subsequent images. In this situation, extremely accurate Euclidean reconstruction is possible as shown by Graham [21]. Nevertheless in this effort and the analysis below, no such assumptions or prior knowledge is known. Instead the techniques below provide solutions to the most generalized and inherently complex data sets, those in which multiple cameras have been used and no prior knowledge exists relating camera positions.

2.4.3 Feature Extraction.

The next step in scene reconstruction is the extraction of invariant image features. Significant research has been conducted in this field resulting in several techniques with the ability to extract hundreds even thousands of keypoints [22, 27]. The most popular techniques entail edge detection algorithms due to their robustness and repeated ability to extract keypoints. Both the Scale Invariant Feature Transform (SIFT) algorithm and the Harris Corner Detector accompanied with the Difference of Gaussian (DoG) filter are used in the software algorithms and will therefore warrant additional attention.

2.4.3.1 Harris Corner Detector.

Invariant image features prove an excellent source for image correspondences due to the minimal dependence upon changes in perspective, scene intensity, and other image characteristics which differ between images. Corners, defined as the intersection of two edges, are outstanding sources since they are rotationally invariant and therefore suitable for the multi-view problem. It is important to note most corner detectors, including the Harris Corner Detector, systematically search images for large omnidirectional gradients which may not correlate to a real world corner. Regardless of this shortcoming, corner detection continues to be a leading method in feature extraction.

To exploit this rotationally invariant feature, Harris and Stephens [22] improved upon existing corner detection algorithms notably Moravec's corner detector [30] by mapping the intensity differential of a test window, $w(x, y)$, which translates over an image with respect to direction. This refinement increased repeatability under small image variations and near edges; however, the detector remains very sensitive to changes in image scale and requires all input images to have identical pixel dimensions.

The filter functions on the premise a small test window records intensity variations as it translates over the surface of the image, and in such case translation over a corner results in a large intensity shift in all directions, as seen in Figure 7

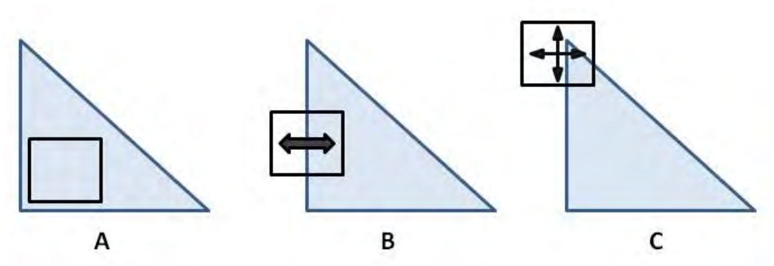


Figure 7. Effects of local window intensity as it translates over a homogeneous region (A), edge (B), and corner (C) of a digital image (blue triangle).

As stated earlier, the Harris filter is based upon Moravec's previous efforts. Moravec's detection filter succinctly determines both edges and corners, but edges are overemphasized, and the response is extremely noisy. The Moravec's filter is mathematically expressed as

$$E\left(u,v\right)=\sum_{x,y}w\left(x,y\right)\left[I\left(x+u,y+v\right)-I\left(x,y\right)\right]^2 \tag{12}$$

where E represents the difference between the original and translated window, u and v are the translation magnitudes in the x and y direction, $w\left(x,y\right)$ represents the location of the window mask, $I\left(x+u,y+v\right)$ is the measured intensity within the translated window, and $I\left(x,y\right)$ is the intensity of the original window.

The principle distinction between Moravec's treatment and Harris and Stephen's is the inclusion of Taylor series expansion of the intensity terms as well as the inclusion of a Gaussian window mask [22]. The difference in intensities is now composed of the first order Taylor expansion accompanied with a Gaussian weighting function as seen in Equation 13

$$E\left(u,v\right) \cong \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ \\ v \end{bmatrix} \tag{13}$$

where M is computed from image derivatives. M in itself is defined as

$$M = \sum w\left(x,y\right) \begin{bmatrix} I_x^2 & I_x I_v \\ \\ I_x I_v & I_v^2 \end{bmatrix} \tag{14}$$

where I_x and I_y are the partial derivatives of $I\left(x,y\right)$ with respect to x and y, and $w\left(x,y\right)$ is the Gaussian weighting function.

Finally, the response of the Harris corner detector is the difference between the determinant of M and the trace squared of M scaled by an empirically determined constant, k

$$\mathcal{R} = det\ \mathrm{M} - k\left(trace\ \mathrm{M}\right)^2 \tag{15}$$

where k typically has values between 0.04 and 0.06 [22].

The value of $\mathcal{R}$ only depends upon the eigenvalues of M and is therefore large for corners yet negative for edges thereby mitigating Moravec's issues with edge detection. In this manner a region is classified as a corner if the response function, $\mathcal{R}$, detects an eight way local maximum. Correspondingly, edges are classified as such if the response is both negative and local minima in either the x or y direction.

2.4.3.2 Difference of Gaussian Filter.

Whereas the Harris corner filter excels at locating corners, the presence of edges and lines are of great importance to epipolar geometry. The Difference of Gaussian filter (DoG) excels in this area and acts as a band pass filter enhancing the visibility of edges and other details within an image while limiting the effects of high frequency noise, a common problem for edge detectors. The DoG filter detects edges within the image by subtracting one blurred version of an image by a second, less blurred version of the original. Mathematically, this is accomplished using a Gaussian filter as seen in Equation 16

$$G_{\sigma_i}(x, y) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{\frac{x^2+y^2}{2\sigma_i^2}} \tag{16}$$

with Gaussian line widths σ_1 and σ_2 where $\sigma_1 > \sigma_2$.

The process begins with obtaining two blurred images, g_1 and g_2.

$$g_1(x, y) = G_{\sigma_1}(x, y) * f(x, y) \tag{17}$$

$$g_2(x, y) = G_{\sigma_2}(x, y) * f(x, y) \tag{18}$$

Finally the difference between the two smoothed images results in the Difference of Gaussian function.

$$g_1(x, y) - g_2(x, y) = (G_{\sigma_1} - G_{\sigma_2}) * f(x, y) = DoG * f(x, y) \tag{19}$$

Figure 8 represents a visual depiction of the DoG filter where the observer can note the inverted Mexican hat profile characteristic of the more general Laplacian of Gaussian filter.

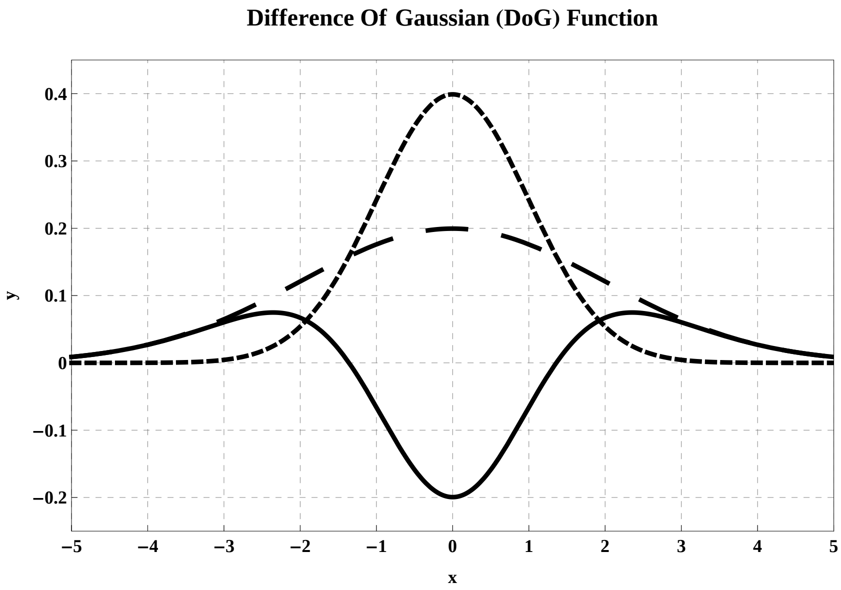


Figure 8. Typical inverted mexican hat seen in DoG filter (solid line). DoG filter resulted from difference of two Gaussian functions with $\sigma_1 = 2$ (dashed) and $\sigma_2 = 1$ (dotted) respectively.

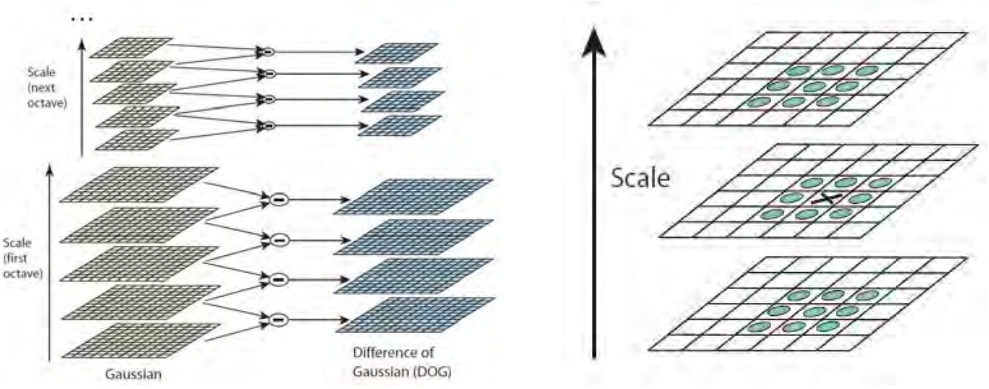
2.4.4 Deriving Image Correspondences: Scale Invariant Feature Transform (SIFT).

The preceding section simplifies the complex problem of registering two images to one where only a discrete number of points must be co-registered. This registration process correlates feature points, x and x', of the same real world point recorded by the two images. The process of matching invariant image features between images, and thereby generating image correspondences, builds the foundations of epipolar geometry and ultimately the success of the reconstruction process.

Several techniques are available to relate extracted feature to one another, and the Scale Invariant Feature Transform (SIFT) operator designed by David Lowe [27] has become the industry standard in image registration due to its ability to identify and register numerous features across large image sets. SIFT operates by detecting keypoints using a cascade filtering approach to first identify candidate locations which are then subjected to further examination. The uniqueness of the cascade filtering approach is best characterized by exploitation of variable scale space where keypoints are found at the inherent image scale but across a variable scale space. This evolution allows SIFT to determine a robust set of scale invariant keypoints.

The SIFT process is best explained by discussing the extraction of keypoints of a single image and then repeating the process to all images contained within the image database. In essence, the SIFT process includes five steps which are described in detail below: scaling each image, convolving it with multiple Gaussian functions, subtracting the Gaussian representations, comparing neighboring differences to extract scale invariant keypoints within the original image, and finally describing each keypoint by its local image gradient.

The first step entails scaling each image into a $n \times n$ subset of the original image to provide the basis for determining scale invariant keypoints. Different octaves are obtained by reducing the scale factor by 2 between each octave to define the various scale spaces. This step ensures only those features which readily occur across multiple scales will be recorded as keypoints. Secondly, each scaled image is convoluted by a variable scale Gaussian function where the standard deviation is varied by $\sigma_{i+1} = \sqrt{2}\sigma_i$ from image to image. This populates each octave with multiple Gaussian convolutions as seen in left side of Figure 9a. A Difference of Gaussian function is applied in the third step to identify potential keypoints similiar to the process described in Subsection 2.4.3.



(a) SIFT octaves [27]

(b) DoG comparison pixels [27]

Figure 9. Representation of the SIFT octaves and calculation of maxima and minima between each octave.

The fourth step extracts the minima and maxima keypoints from the filtered images by comparing a pixel to its 26 neighbors in a 3-by-3 region at the current and adjacent scales as seen in Figure 9b which is a detailed representation of the right side of Figure 9a. The pixel is selected if and only if it is a local maxima or minima when compared to adjacent pixels. Lowe states the cost of checking all pixels and comparing each to all neighbors is reasonably low due to the fact most sample points are eliminated within the first couple of iterations.

The fifth and final step computes the keypoint descriptor as seen in Figure 10. To compute the keypoint descriptor, SIFT maps the gradient and orientation of each image point surrounding the keypoint locations using a Gaussian window to weigh neighbor contributions. For purposes of this explanation, a 8-by-8 region falling within the Gaussian window constitutes a 4-by-4 subregion where the individual gradient magnitude is added to the nearest bin.

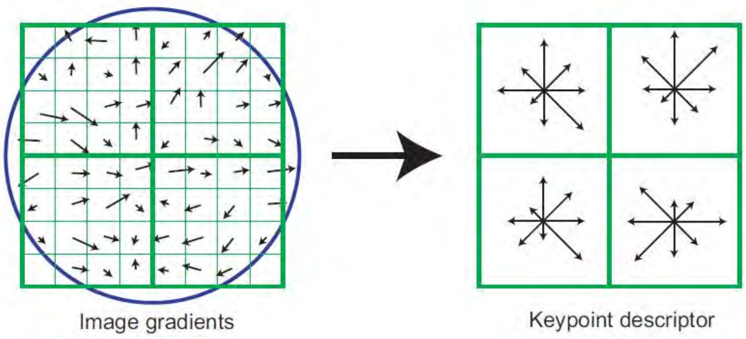


Figure 10. Generation of keypoint descriptors [27].

In reality the algorithm uses a 16-by-16 region for comparison. The mapping is stored in a $4 \times 4 \times 8 = 128$ element feature vector for each keypoint which can be used to compare against the regional descriptions of keypoints in other images which follow an identical process as that described above. Those image keypoints falling within the closest histogram vectors are assigned as potential matches and referred to as nearest neighbors. Lowe continued to show the keypoints are particularly invariant to image rotation and scale, robust across variable affine distortions, noise, and illumination differences which make this method of feature detection suitable for the purposes of this effort.

The output results of the SIFT algorithm are shown in Figure 11. In each image roughly 1000 keypoints were generated from a fairly featureless scene which highlights the applicability of the SIFT algorithm. It is important to note that not all image keypoints will be matched to keypoints from another image as seen in right side of Figure 11. This is due to a number of issues including keypoint obscuration from sequential images which stresses the importance of image selection. In principle, the camera focal points must be sufficiently distinguished from one another yet not to such an extent to hinder matching of sufficient number of image correspondences.

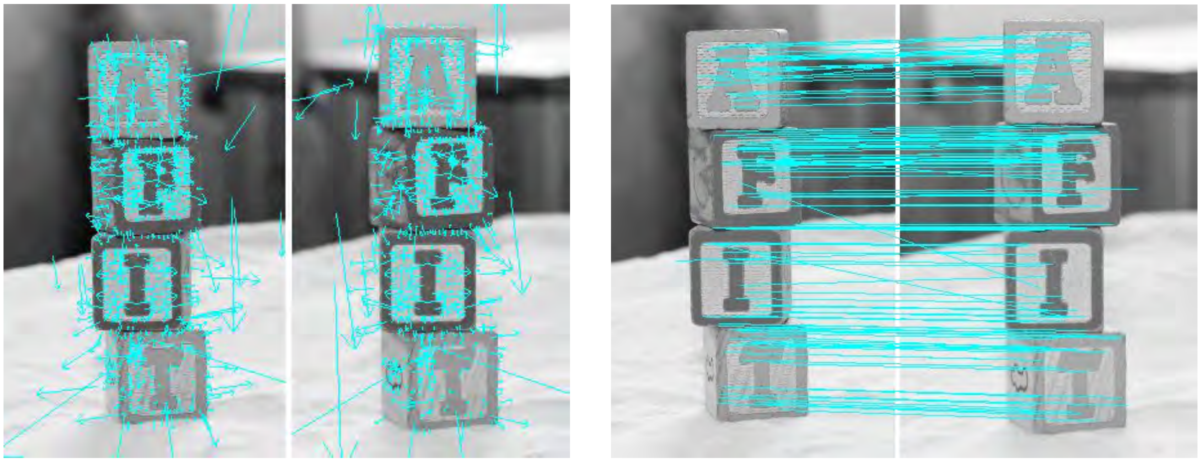


Figure 11. Scale Invariant Feature Transform results. In the left image, the blue arrows represent feature locations and magnitude of gradient. Note the images contain 1026 and 927 keypoints. The right images portrays the matching results based upon the gradient vectors. In total, 134 matches were found between the images with only two outliers.

2.4.5 Derivation of the Fundamental Matrix.

In deriving the fundamental matrix, F, we must return to the epipolar geometry described in Section 2.3 with the addition of correspondences determined by SIFT represented by $x_1 \mapsto x'_1$ and $x_2 \mapsto x'_2$ in Figure 12. The fundamental matrix states for a real world point X existing on plane π the ray must pass through the camera center and appear as point x on the focal plane of the camera. When viewed from the second camera center, ray $\overline{X_1 C}$ will map to the second camera's focal plane as l'_1.

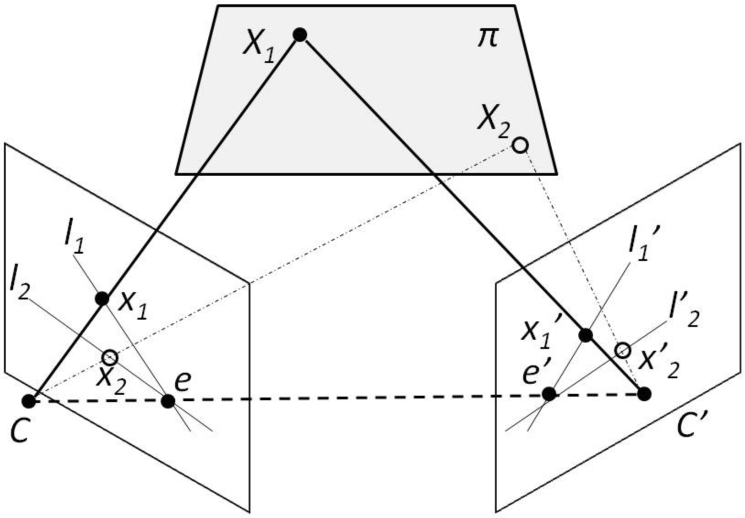


Figure 12. Epipolar geometry with two keypoint correspondences [23].

The fundamental matrix is the principle algebraic representation of the epipolar geometry described in Figure 12. Each point x maps to an epipolar line, l', according to $x \mapsto l'$. This correlation represents a mapping from points to lines which represents the fundamental matrix. Mathematically the fundamental matrix is a 3×3 matrix of rank 2 containing seven degrees of freedom if the constraint det F $= 0$ is enforced.

The fundamental matrix can be derived in a number of manners; however, a geometric derivation is provided. Referring to Figure 5, x relates to x' through a 2-D homography matrix, H_π, where H_π represents the transfer mapping from one image to another via plane π. Therefore $x' = H_\pi x$ and the epipolar line may be written as $l' = e' \times x'$. Following the substitution, $x' = H_\pi x$, we arrive at the fundamental matrix definition.

$$l' = e' \times H_\pi x = Fx \tag{20}$$

The most basic property of the fundamental matrix is seen in Equation 21 which is required between any two image correspondences,

$$x'^T F x = 0 \tag{21}$$

thereby relating and constraining SIFT derived correspondences, $x_i \mapsto x_i'$, found in the previous section.

In computing F, each correspondence is represented in homogenous notation as $x = [x, y, 1]^T$ and $x' = [x', y', 1]^T$ and seven such correspondences are required to provide one linear equation for each unknown entry. If F consists of a 3×3 matrix with each entry represented by f_{ij}, then Equation 21 can be used to write the equations in the following form,

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0 \tag{22}$$

If this equation is written as a vector inner product of the 9-element vector, f, the following form can be derived.

$$\begin{bmatrix} x'x & x'y & x' & y'x & y'y & y' & x & y & 1 \end{bmatrix} f = 0 \tag{23}$$

Finally for a set of n matches a homogenous set of linear equations where $x'^T F x = 0$ determines a set of equations in the form $Af = 0$.

$$Af = \begin{bmatrix} x_1'x_1 & x_1'y_1 & x_1' & y_1'x_1 & y_1'y_1 & y_1' & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n'x_n & x_n'y_n & x_n' & y_n'x_n & y_n'y_n & y_n' & x_n & y_n & \end{bmatrix} f = 0 \tag{24}$$

The solution to matrix A in Equation 24 must have at most a rank of 8 to be unique and found using linear methods such as the direct linear transform. However in the presence of noise or when the matrix is overdetermined, the rank of matrix A may be 9 and demand a least-squares solution. In such case a non-linear minimization technique such as the Levenberg-Marquart method is preferred. Calculations in the form of $Ax = 0$ where A is known are extremely common in SfM/MVS computations. For example, the fundamental matrix, epipolar lines, and triangulation of the 3-D coordinates as will shortly be discussed all occur in this format. Therefore it is common practice to find x that minimizes $\|Ax\|$ subject to $\|x\| = 1$ by computing the SVD of A where $A = UDV^T$ and x is the last column of V.

2.4.6 Derivation of Camera Projection Matrices, Π, from F.

The camera projection matrices, Π_1 and Π_2 for a two-view system, relate the real world coordinates to the actual image coordinates. Hartley and Zisserman determined if and only if $\Pi_2^T F \Pi_1$ is skew symmetric, the fundamental matrix corresponds to the projection matrices, Π_1 and Π_2. If we invoke projective ambiguity Π_1 can be simply defined as

$$\Pi_{1p} = [I|0] \tag{25}$$

where I is a 3×3 identity matrix. The second projection matrix involves the skew symmetric matrix defined as,

$$[e']_\times = \begin{bmatrix} 0 & -e'_3 & e'_2 \\ e'_3 & 0 & -e'_1 \\ -e'_2 & e'_1 & 0 \end{bmatrix} \tag{26}$$

where e' can be directly related to the translation, T, between the camera centers when known or the relationship $F^T e' = 0$. Therefore the second projection matrix can be defined as

$$\Pi_{2p} = \left[[e']_\times F | e'\right] \tag{27}$$

2.4.7 Initial Estimate of 3-D Coordinates.

The initial estimate of the 3-D scene is accomplished using the camera matrices defined in the previous section and linear triangulation methods. The method of linear triangulation is directly related to the direct linear transform or SVD methods used to solve for the fundamental matrix. Therefore when $x = \Pi_{1p}X$ and $x' = \Pi_{2p}X$ are combined into $AX = 0$ the linear set of equations in X is developed.

$$\begin{aligned} x \times (\Pi_{1p}X) &= 0 \\ x\left(\Pi^{3T}\right) - \left(\Pi^{1T}\right) &= 0 \\ y\left(\Pi^{3T}\right) - \left(\Pi^{2T}\right) &= 0 \\ x\left(\Pi^{2T}\right) - y\left(\Pi^{1T}\right) &= 0 \end{aligned} \tag{28}$$

where Π^{iT} are the rows of Π. Finally A can be represented as

$$A = \begin{bmatrix} x\Pi_{1p}^{3T} - \Pi_{1p}^{1T} \\ y\Pi_{1p}^{3T} - \Pi_{1p}^{2T} \\ x'\Pi_{2p}^{3T} - \Pi_{2p}^{1T} \\ y'\Pi_{2p}^{3T} - \Pi_{2p}^{2T} \end{bmatrix} \tag{29}$$

Figure 13 shows the sparse point cloud and camera poses derived from a set of 13 images related to those found in Figure 11 resulting in 1539 vertices. This illustration accurately depicts SfM's ability to not only accurately reconstruct the scene but also identify relation pose of cameras which is known through the relationship, $C = -R' * T$ from the provided R and T motion components of the rigid body estimate.

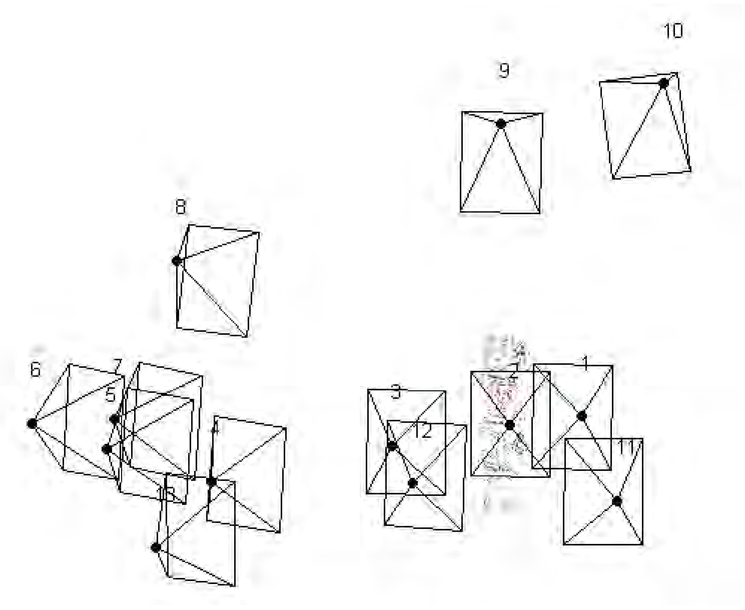


Figure 13. Structure from Motion reconstruction of AFIT blocks. Scene reconstructed with 13 cameras resulting in 1539 vertices.

2.5 Random Sampling Consensus (RANSAC)

Unfortunately real world data contain erroneous values which negatively influence the final reconstruction. Therefore a method to remove outliers is preferable and of great benefit to any linear or non-linear process. Typically a least squares approach to estimating parameters involves analysis of all data points under the assumption all points are valid. However when data points do not fall within this assumption, erroneous points lead to misrepresentation of the ideal model line. Random Sampling Consensus (RANSAC), a robust parameter estimator developed by Fischler and Bolles, iteratively removes gross outliers to mitigate their effect on the final solution [17]. Typically within the computer vision problem, erroneous correspondences lead to poor estimation of the fundamental matrix, and RANSAC alleviates these measurement errors by culling the gross outliers. Optimization after outlier removal leads to a much more usable final solution. Figure 14 shows several well-behaved points and a gross outlier. Standard least squares regression produces a poor fit to the data, whereas removal of the gross outlier results in an accurate linear fit.

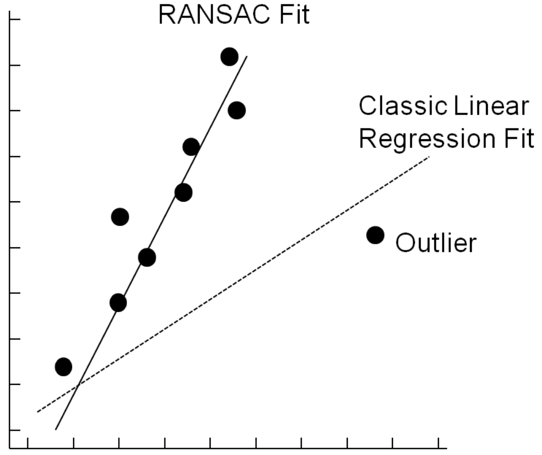


Figure 14. RANSAC line fitting compared to iterative least squares.

The RANSAC process fits a line to a set of randomly selected data points with a minimally derived model containing two points. All data points are then analyzed to determine the number of inliers existing a certain threshold distance from the fitted line. The process is repeated a statistically relevant number of times and the model, or fitted line, which contains the most inliers is kept.

2.6 Sparse Bundle Adjustment

Bundle Adjustment (BA) is typically the final step in any feature-based 3-D reconstruction algorithm and involves refining the 3-D structure and camera parameters from the initial estimates through the use of a non-linear minimization algorithm such as the Levenberg-Marquardt technique. The Sparse Bundle Adjustment (SBA) techniques developed by Lourakis and Argyros [25, 26] iteratively minimizing the reprojection error between the observed and predicted images points which is expressed by the sum of the squares of a multitude of nonlinear functions. Specifically, the term bundle adjustment refers to the simultaneous estimate of multiple camera parameters including their relative motion and pose, the reconstructed points, and the bundle of rays emanating from the vertex which converge at the camera center. Inherently, the equations used to solve this term are sparse due to the lack of integration between the 3-D points and the cameras. Many reconstructed vertices are only viewed in a small subset of images and even fewer images are used to calculate the pixel depth. Therefore, the interaction between the camera parameters and reconstructed vertices is inherently sparse. The reader will quickly gain an appreciation for the sparseness of the hessian matrix which must be solved during the Levenberg-Marquardt routine as seen in Figure 15 for a relatively simplistic model consisting of three cameras and four points.

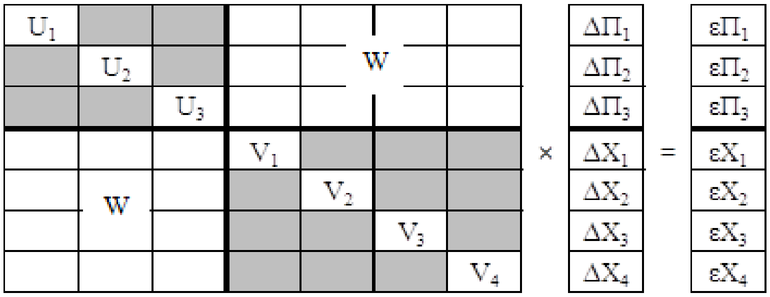


Figure 15. Form of normal equations in a hessian matrix.

where matrix elements are defined below [23].

$$U_i = \sum_i \left(\frac{\delta x_{ij}}{\delta \Pi_j}\right)^T \Sigma_{x_{ij}}^{-1} \left(\frac{\delta x_{ij}}{\delta \Pi_j}\right) \tag{30}$$

$$V_i = \sum_j \left(\frac{\delta x_{ij}}{\delta X_j}\right)^T \Sigma_{x_{ij}}^{-1} \left(\frac{\delta x_{ij}}{\delta X_j}\right) \tag{31}$$

$$W_{ij} = \left(\frac{\delta x_{ij}}{\delta \Pi_j}\right)^T \Sigma_{x_{ij}}^{-1} \left(\frac{\delta x_{ij}}{\delta X_i}\right) \tag{32}$$

$$\epsilon_{\Pi_j} = \sum_i \left(\frac{\delta x_{ij}}{\delta \Pi_j}\right)^T \Sigma_{x_{ij}}^{-1} \epsilon_{ij} \tag{33}$$

$$\epsilon_{X_j} = \sum_i \left(\frac{\delta x_{ij}}{\delta X_j}\right)^T \Sigma_{x_{ij}}^{-1} \epsilon_{ij} \tag{34}$$

$$\Delta P_j = P_j - P_{j-1} \tag{35}$$

$$\Delta X_i = X_i - X_{i-1} \tag{36}$$

As this model is expanded to larger numbers of vertices and cameras as found in this research the sparseness of the near-Hessian matrix is evident as seen in Figure 16.

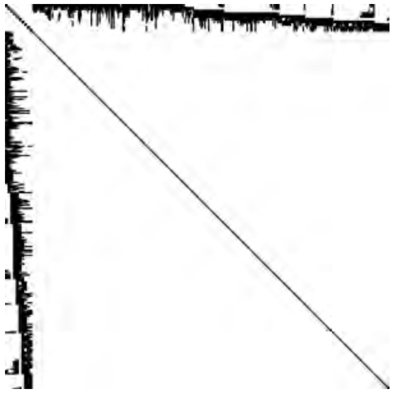


Figure 16. Sparse structure common to 3-D reconstruction matrices[26]. Note non-zero elements are black.

2.7 Dense Point Cloud Reconstruction

To this point the SfM techniques described above produce a sparse reconstruction of the scene without prior scene knowledge, use of calibrated cameras, or structured image collection processes. However, Multi-View Stereo (MVS) algorithms incorporate camera motion found during the SfM reconstruction process to derive a dense reconstruction. There are four approaches to computing a denser point cloud which include voxel based, deformable polygonal meshes, multiple depth maps, and patch-based methods [19]. The purpose of this thesis is to investigate the reconstruction of a complex urban environment with substantial building and vegetative obscuration. Under these conditions voxel and deformable polygonal meshes reconstruction techniques are extremely limited and multiple depth map techniques become exponentially more complex as the number of input images exceeds three. Therefore, the patch-based method is ideally suited for reconstruction of urban environments and scales well with multiple images.

2.7.1 Patch-Based Multi-View Stereo Algorithm.

The patch-based MVS (PMVS) algorithm [11, 19], developed by Furukawa and Ponce in 2008, produces a dense point cloud consisting of patches, or vertices, describing the scene based on correspondences between image sets similar to the SfM methods described above. PMVS leverages knowledge gained from the sparse point cloud formation to further increase the number of correspondences between images by constraining the correspondence search to points along similar epipolar lines attaining a nearly pixel level reconstruction. This patch-based MVS algorithm consists of three steps: matching, expanding, and filtering.

Initial feature detection is performed using Harris Corner Detector and Difference of Gaussian filters in parallel as described in Subsection 2.4.3. As opposed to sequentially searching each images in its entity, each image is subdivided into a rectangular search grids consisting of 32-by-32 pixel blocks. The top four maxima determined by each feature detector, either Harris or DoG, are retained for each block. This step ensures features are sampled evenly across the scene where feature rich areas are sampled similarly to homogeneous or feature deficient areas to ensure completeness of the resulting dense point cloud.

With the image features, denoted by f, in hand, the algorithm collects the features that lie within two pixels of the corresponding epipolar lines. From these correspondences, 3-D points are triangulated from the feature pairs, (f, f'). These 3-D points are considered potential patch centers, p, where the patch is defined by its center $c\left(p\right)$, normal vector $n\left(p\right)$, and a reference image $R\left(p\right)$ in which p is visible. Furukawa summed these descriptors in the following equations

$$\begin{aligned} c\left(p\right) &\leftarrow \{\text{Triangulation from f and f'}\}, \\ n\left(p\right) &\leftarrow \frac{c\left(p\right)O\left(I_i\right)}{|c\left(p\right)O\left(I_i\right)|}, \\ R\left(p\right) &\leftarrow I_i \end{aligned} \tag{37}$$

where $O\left(I_i\right)$ is the optical center of the corresponding camera. The set of images in which the patch is visible is defined as,

$$V\left(p\right) \leftarrow \{I|n\left(p\right) \cdot \overrightarrow{c\left(p\right)O\left(I\right)} \backslash |\overrightarrow{c\left(p\right)O\left(I\right)}| > \cos\left(\iota\right)\} \tag{38}$$

where ι is the angle between the patch normal and direction of the patch to the optical center of the camera and typically defined at $\frac{\pi}{3}$. At this point a set of filtered images $V^*\left(p\right)$,

$$V^*\left(p\right) = \{I|I \in V\left(p\right), h\left(p, I, R\left(p\right)\right) \leq \alpha\} \tag{39}$$

based on $V\left(p\right)$ is computed to ensure each image meets a pairwise photometric discrepancy score with a reference image selected from $V\left(p\right)$. The photometric discrepancy score is described below

$$g\left(p\right) = \frac{1}{|V^*\left(p\right)\backslash R\left(p\right)|} \sum_{I \in V^*\left(p\right)\backslash R\left(p\right)} h\left(p, I, R\left(p\right)\right) \tag{40}$$

where $h(p, I, R(p))$ is simply the pairwise photometric discrepancy function between two images. When the potential matches are filtered to eliminate those with poor photometric scores and $V^*(p)$ is updated $g(p)$ is similarly updated to $g^*(p)$ with the substitution $V(p) \rightarrow V^*(p)$. This subset of filtered images ensures anomalous features such as specular reflections are omitted. $V^*(p)$ is optimized by minimizing $g^*(p)$ which confines $c(p)$ to remain on the ray passing from its projection on the visible image to the camera center. In this manner, optimization is constrained to depth alone. If the patch is found in a predefined number of images, i.e. $V^*(p) > \gamma$, the patch creation is deemed successful, and p is recorded in the appropriate image cell.

During the expansion step, it is desired to construct one patch for each image cell. Furukawa and Ponce accomplished this by seeding new patches with existing ones under certain constraints. The first constraint is quite elementary in that if a cell contains a patch no expansion is necessary. The second constraint concerns depth discontinuities which exist when an image cell from one camera records a feature close to the camera but the corresponding camera records a different depth. In such case, the patch is deemed erroneous and rejected. New patch candidates are created from the neighboring patch parameters $n(p)$, $R(p)$, and $V(p)$, into $n(p')$, $R(p')$, and $V(p')$. $c(p')$ is then determined as the intersection point between the ray passing from the center of the new image cell through the plane defined by p. Further expansion steps mirror those described in the previous section in which Equation 39 is used to determine $V^*(p)$ while $c(p)$ and $n(p)$ are optimized.

Patch formation and expansion is typically well behaved, but erroneous patches will occur. In the final PMVS step, three filters are used to remove these erroneous patches relying on visibility consistency and regularization. The first filter removes patches which are not neighbors but are stored within the same cell. The second filter also requires visibility consistency where the number of images contained in $V^*(p)$ exceeds γ. Finally the third test enforces regularization. Each patch must have a proportion of neighbors in identical and adjacent cells in all images. This final filter can be thought of as requiring each patch to have a nearest neighbor within a certain distance and those patches without neighbors are isolated and therefore removed.

2.7.2 Clustering Views for Multi-View Stereo.

Agarwal and Furukawa et. al. developed the techniques to rapidly calculate scene geometry from hundreds and even thousands of images by clustering images into manageable sized clusters and independently calculating the scene geometry using the techniques described in Subsection 2.7.1. The technique appropriately named clustering multi-view stereo (CMVS) [13, 5] uses information from the SfM processes to cluster images for sequential or parallel processing using MVS techniques. The CMVS clustering algorithm consists of four steps: merging SfM points, remove redundant images, enforce size constraints, and finally enforce coverage constraints [18].

The first step reduces the number of SfM points to improve processing time of the remaining steps. An SfM point is randomly selected and merged with neighboring points by aggregating visibility data over the local neighborhood. The randomly chosen point and all those merged are removed from the data set. This merging process continues until the data set is empty. The second step removes redundant images by testing each image independently and removing it if a coverage constraint still holds. This culling process is continued until the visibility constraint fails or the number of images per SfM point is reduced to three. If removal of a particular image does not violate either of these constraints the image is permanently discarded. The third step ensures image clusters do not violate a size constraint typically related to the available memory without regard to coverage. If the cluster is larger than a user defined maximum size, the cluster is divided into smaller clusters by weighing images which have a high MVS contribution. The final step enforces coverage which was ignored in the previous step. In this step images are added to the clusters to cover additional SfM points. A series of prioritized steps are constructed which add specific images to the cluster and record the effectiveness of image addition. Initialized with the step of highest priority, the step is executed and deemed successful if the new coverage exceeds 0.7 times the highest score. In such case the image cluster is updated to include the new image. This process is repeated until the coverage constraint and cluster size constraints are satisfied [18].

The final product of the SfM and MVS pipelines is demonstrated in Figure 17. The reconstruction contains 10935 vertices representing over a seven fold increase in the number of reconstructed 3-D points. The substantial increase in vertices produces a noticeably denser reconstruction.

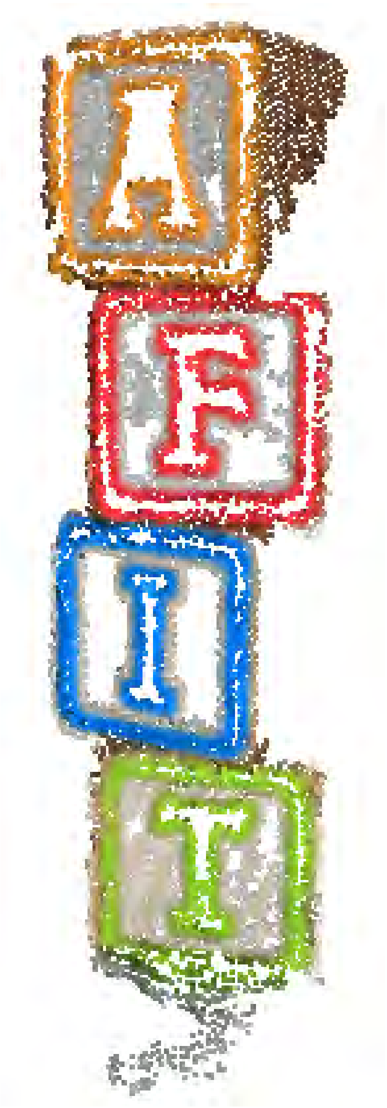


Figure 17. Dense reconstruction of AFIT blocks. Scene reconstructed with thirteen cameras resulting in 10935 vertices.

The depth information contained within the above reconstruction can be visualized through the optical process, stereopsis, which was previously discussed. The stereopically derived 3-D effect can be visualized by crossing the eyes until the two images in Figure 18 merge. Each eye focuses independently on the left and right images while the brain combines the images to relay the three dimensional object.

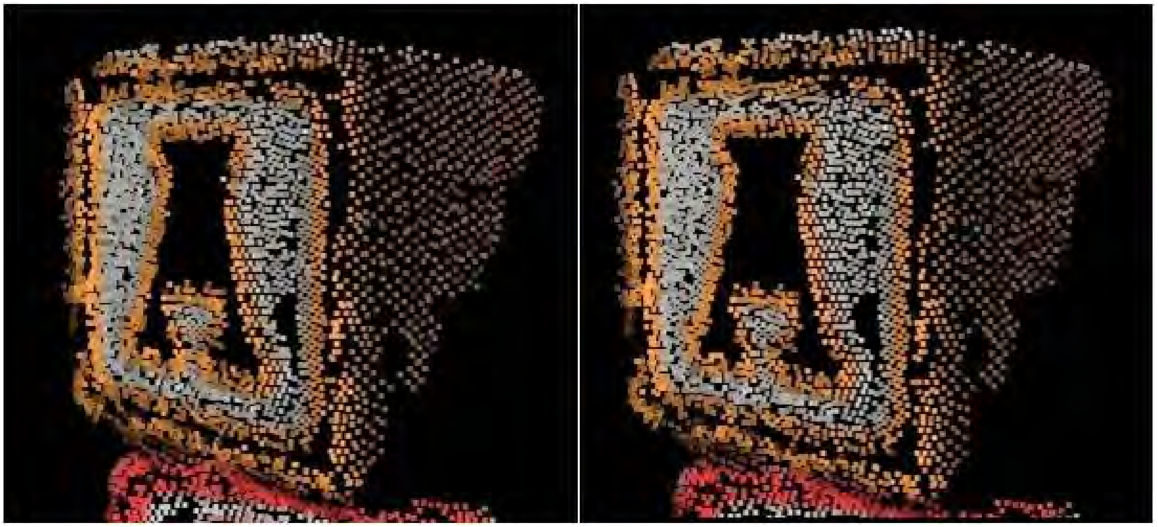
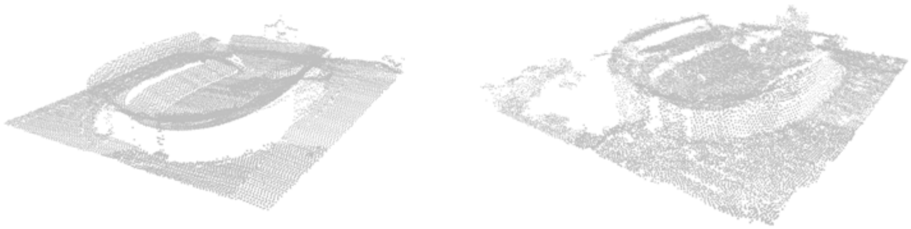


Figure 18. Stereopsis effects reveal the depth information contained within the dense reconstruction.

2.8 Case Study: Reconstruction of the Ohio State University Stadium

The Ohio State University stadium was reconstructed using fifty images extracted from the Columbus Large Image Format (CLIF) II October 2007 data set provided by AFRL/RY. This data set presents a unique opportunity to compare the dense reconstruction to LIDAR data collected with a Leica ALS50 digital LIDAR system with average post spacing of seven feet. The LIDAR data was sourced from the Ohio Geographically Reference Information Program [10]. Figure 19 represents the capability of the Bundler/PMVS workflow to generate a high quality point cloud representation of the target area in direct comparison to the LIDAR data.



(a) LIDAR (b) Image Derived

Figure 19. LIDAR derived point cloud compared to an image derived reconstruction.

Figure 19 highlights several strengths and weaknesses of the image derived and LIDAR 3-D reconstructions. First LIDAR presents very little noise in the vertical direction; however, since ground sampling is equally distributed, structure edges are left unresolved. This significantly hinders the reconstruction of vertical structures with ground dimensions less than the LIDAR sampling distance, typically 5 meters from an altitude of 10000 feet. On the other hand, image derived reconstructions derive vertices from intensity gradients of which edges are primary contributors. For instance, image derived reconstruction can reconstruct vertices on vertical walls while retaining color information where LIDAR cannot. Finally since image features are required to relate images, image derived reconstructions fail in areas of homogeneous and specular surfaces where LIDAR continues to provide a return. When noting the depth accuracy inherent to LIDAR and spatial x and y accuracy of image reconstruction, it is easily imagined fusion of LIDAR and image derived reconstructions will be a ripe area for further research.

2.9 Reconstruction Ambiguity

The mathematical techniques discussed above result in projective reconstruction yet other variations exist resulting in a degree of reconstruction ambiguity. Each reconstruction, projective, affine, euclidean, and metric, is distinguished by the gradual insertion of in-situ knowledge into the final solution and therefore represent improved reconstructions of the scene. Projective reconstruction methods, denoted as Pi_p in the mathematics above, serve as the most basic but generalized method which require no foreknowledge of the original scene. These reconstructions are characterized by the failure to preserve parallel lines and metric distances which result in significant skewing of the original 3-D scene.

Affine reconstructions represent the first upgrade in which parallel lines and right angles are preserved. Typically when imaging man-made structures one deductively infers basic geometric shapes such as rectangles and cylinders. With this knowledge it is possible to compute the vanishing point defined as the point in which parallel lines in a projective reconstruction appear to converge. This principle is clearly seen in the convergence of distant railroad tracks whereas in reality the tracks are parallel. Computation of the vanishing point corrects the projective reconstruction, but relative distances between 3-D points is not maintained. In other words parallel lines are preserved but each axis is reconstructed to different scale factors.

Euclidean reconstructions offer substantial improvements to the projective and affine reconstructions, yet require autocalibration steps or the foreknowledge of the camera calibration matrix to provide partially or fully calibrated camera frames. The visual renditions of sample reconstructions seen throughout this effort are of euclidean nature unless otherwise noted, and all employed basic autocalibration measures. This step is frequently skipped since the Euclidean reconstruction can be determined from the perspective reconstruction if and only if knowledge of the camera calibration matrix, K, is known. However, Ma Yi, et.al. have shown by using the absolute quadric constraint and assuming K is constant between all images, a final euclidean reconstruction can be achieved [28].

Finally the ultimate goal of true metric reconstructions require knowledge of the camera's true IOPs and EOPs. Thus far the inclusion of the camera's IOPs and EOPs has been omitted from the SfM and MVS processes, and only relative R and T motions included. This was done to provide a generalized reconstruction problem applicable to all data sets. Typically the camera's EOPs are transmitted in the form of yaw, pitch, and roll of the sensor as well as the latitude, longitude, and altitude of the camera system. Simple conversions relate this real world information to the R and T components necessary to derive an absolute world coordinate system model in which point to point distances and model translation and rotations are all true to a real world coordinate system.

2.10 Autocalibration

A final theoretical construct which remains to be discussed is autocalibration. Autocalibration techniques provide the homography matrix required to upgrade the projective reconstructions to euclidean without complete knowledge of the camera's IOPs/EOPs. The following steps are outlined in greater detail by Ma [28], but are surmised here.

The projective reconstruction X_p is related to the euclidean reconstruction X_3 by a homography matrix through, $X_p \sim HX_e$ where

$$H = \begin{bmatrix} K_1 & 0 \\ -v^T K_1 & 1 \end{bmatrix} \tag{41}$$

K_1 is the camera calibration matrix for the first camera, and v^T represents the solution to the linear constraints seen below. At this point it is important to distinguish R_i and T_i from R and T. Whereas the latter represent the rotation and translation with respect to the rigid body motion, R_i and T_i denote the euclidean motion from the first camera frame. The camera projection matrices are thereby related by Equation 42.

$$\Pi_{ip} H \sim \Pi_{ie} = [K_i R_i, K_i T_i] \tag{42}$$

Furthermore, the homography matrix is restricted to the left 3×3 block since the far right column fails to constrain H. R_i may be eliminated by multiplication of K_1^T resulting in the absolute quadric constraint center matrix seen below.

$$\Pi_{ip} Q \Pi_{ip}^T \sim S_i^{-1} \tag{43}$$

where $S_i = K_i^{-T} K_i^{-1}$ and Q is defined as

$$Q = \begin{bmatrix} K_1 K_1^T & -K_1 K_1^T v \\ -v^T K_1 K_1^T & v^T K_1 K_1^T v \end{bmatrix} \in \Re^{4\times4} \tag{44}$$

Fortunately this problem is greatly simplified under three assumptions: the optical axis is orthogonal to and intersects the center of the imaging plane and the imaging pixels are square. This assumptions lead to the adaption of Equation 43 to

$$\Pi_{ip} \begin{bmatrix} a_1 & 0 & 0 & a_2 \\ 0 & a_1 & 0 & a_3 \\ 0 & 0 & 1 & a_4 \\ a_2 & a_3 & a_4 & a_5 \end{bmatrix} \Pi_{ip}^T \sim \begin{bmatrix} f_i^2 & 0 & 0 \\ 0 & f_i^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{45}$$

resulting the following constraints where π_i^j represent the rows (j) of the camera projection matrix Π for each camera (i).

$$\begin{aligned}
&\pi_i^{1T} Q \pi_i^1 = \pi_i^{2T} Q \pi_i^2 \\
&\pi_i^{1T} Q \pi_i^2 = 0 \\
&\pi_i^{1T} Q \pi_i^3 = 0 \\
&\pi_i^{2T} Q \pi_i^3 = 0
\end{aligned} \tag{46}$$

When Q is in the form of Equation 45 only three camera frames are required and the five unknowns are recovered linearly. Finally K and v are extracted from Q by

$$K_1 = \begin{bmatrix} \sqrt{a_1} & 0 & 0 \\ 0 & \sqrt{a_1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{47}$$

$$v = -\left[\frac{a_2}{a_1}, \frac{a_3}{a_1}, a_4\right]^T \tag{48}$$

which finalizes the unknowns required for the H euclidean upgrade.

2.11 Summary

Structure from Motion provides the tools necessary to reconstruct accurate 3-D scene reconstruction from 2-D images. At the heart of the process is the fundamental matrix which defines $x'^T F x = 0$ and ultimately, with enough correspondences, depth information can be derived from the image correspondences alone. From the image correspondences, a fundamental matrix can be derived which leads to the camera projection matrices and finally the initial 3-D coordinates. Sparse bundle adjustment techniques simultaneously refine the 3-D points and camera centers resulting in a sparse point cloud. Further processing of the data transforms the sparse point cloud to a dense point cloud through a secondary search of additional correspondences by limiting the search over a localized region corresponding to the epipolar lines. However with image correspondences alone, the techniques above are limited to a projective reconstruction. Augmenting the problem with in-situ knowledge such as the camera calibration matrix, K, rotation and translation vectors between cameras, R and T, a direct substitution can be made for the projection matrices and a final Euclidean reconstruction achieved.

III. Methodology

3.1 Chapter Overview

This chapter discusses the software, analytical tools, and test methodology employed to analyze the effects of various collection parameters on the final 3-D reconstruction. The reconstruction techniques discussed in the previous chapter have the ability to accurately reconstruct a diverse set of structures from simple objects to elaborate urban environments. However, past researchers utilized data sets including thousands of images from highly diversified viewing geometries or laboratory environments where all scene angles can be orchestrated in minute detail. Both these scenarios depict rich collection geometries which provide an abundance of information to reconstruct accurate dense point clouds but such collections may be unattainable in certain situations. Modern intelligence gathering scenarios often prohibit the ability to collect data sets of this nature since operational considerations restrict the optimal placement of camera centers and forbid multiple overflights or persistent surveillance. In essence, the collector is limited to a single pass without opportunity of reengagement. Under these qualifying conditions, thorough knowledge of the collection geometry effects on the final reconstruction accuracy is required to ensure a highly accurate and complete reconstruction can be derived from a minima of data. Unfortunately, investigative efforts examining 3-D reconstructions under the restrictions mentioned above are found wanting.

For these reasons, three parameters, number of cameras, convergence angle, and variable viewing geometries were investigated in detail using both a simulated MATLAB environment and a 3-D rendering software capable of generating images of real world environments. First, the MATLAB environment affords the ability to investigate the SfM/MVS algorithm's ability to reconstruct a simple shape and provide a high fidelity trade-off analysis examining the dependence of reconstruction error on the number of camera frames and convergence angles. Secondly, Blender, a 3-D computer graphics software package, enabled the capture of aerial images of a synthetic urban environment modeled after Sadr City, Iraq to simulate operationally realistic collection events. Multiple image data sets were acquired simulating three characteristic reconnaissance flight geometries of an unmanned aerial vehicle including direct linear passes, circular orbits, and finally a hybrid s-curve path.

3.2 Phase One: Fundamental Reconstruction Simulations

The SfM/MVS algorithms were implemented within the MATLAB operating environment for dual purposes. First, the code offers a demonstrative capability to highlight the peculiarities of 3-D scene reconstruction when certain information is withheld. Additionally, the code provides an open source opportunity to manipulate and edit the algorithm which is an attribute not currently available in the academic software packages, Bundler and CMVS/PMVS2, which provide the 3-D reconstructions for the aerial profile data sets. Secondly and serving as the primary purpose, the code allows a precise investigation of the effects of limited convergence angles and number of camera frames on reconstruction accuracy. To support this reconstruction effort, a simple nine point geometric shape was created as seen in Figure 20. It is important to note, the elementary wireframe structure seen in the figure is defined by nine points only. The connecting lines have been drawn in for visualization purposes only and not used by the SfM/MVS algorithms. This arrangement allows for the precise positioning of cameras and full control of translation and rotation. Furthermore, multiple cameras may be added to simulate the effects of numerous viewing geometries and camera poses. This level of control is not without limitation, two of which demand attention.

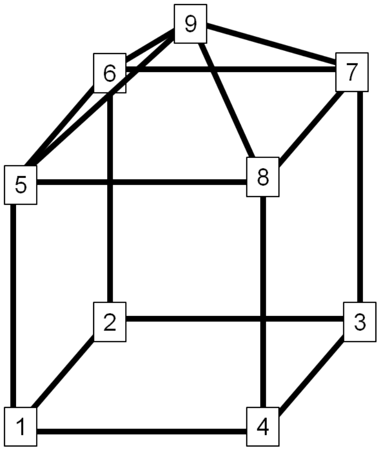


Figure 20. Basic geometric shape for MATLAB reconstruction algorithms. Note lack of opaque surfaces allows for viewing of all features.

3.2.1 Limitations.

The first limitation to this methodology is the inability to occlude points. In solid objects, irradiance from hidden points on the farside of the structure would be occluded from the camera as the rays would be denied a direct transmission path to the camera. Additionally, even if reflective surfaces were introduced, the reflection itself would be identified as a unique feature separate from its originating source. Since the algorithm contains knowledge of all points unhindered by occlusions, extremely accurate reconstructions may be recovered; however, it must be noted this situation exists within the computer environment alone as real world occlusions will degrade reconstruction accuracy considerably. Nevertheless, this limitation allows for verification of the reconstruction code and reveals general trends in the relationship between the collection parameters and reconstruction accuracy. A second limitation to this method is knowledge of perfect correspondences. Within the code, the nine points are recorded as features and directly matched to one another to form the required correspondences. In other words, this limitation allows for the algorithm to have complete knowledge of all points at all times. A side effect of the second limitation is the false apparency of resolution invariance. As the resolution of the images increases the raw image data available for reconstruction increases and therefore reconstruction accuracy is affected. However with the knowledge of perfect correspondences, the reconstruction accuracy is unaffected by the image resolution. For instance, the reconstructions were carried out with multiple image resolutions ranging from 250,000 to 25,000,000 pixels per image with negligible effect on accuracy.

Real world reconstruction algorithms require feature extraction and matching from images in which occlusions, variable resolution, and erroneous correspondences result. In such case, the extraction and matching processes may fail to identify and match all points amplifying the sparse nature of the reconstruction matrices not captured in this fundamental reconstruction simulation.

3.2.2 Procedure.

The effects of convergence angle and number of camera frames on reconstruction accuracy were examined with the MATLAB simulation code. The nine point structure was imaged by translating a camera along a preprogrammed track while automatically rotating each camera to ensure the structure remained in the center of each image. At each point, the 2-D location of the nine points was recorded, thereby producing the required correspondences, x and x'. The convergence angle was defined as the angle formed by the rays originating from the center of the structure to the two most extreme camera locations as seen in Figure 21. Data sets defined by convergence angles from 1° to 100° in 1° increments were recorded with two cameras. Furthermore, at each angle additional camera frames were acquired at equal intervals along a line formed by two extreme camera frames beginning with two and iteratively increasing to twenty. This provides a high fidelity data set consisting of 1900 unique data sets containing all permutations of convergence angles and number of camera frames.

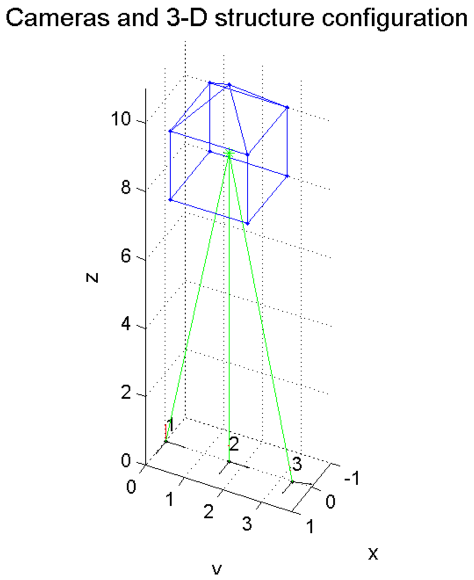


Figure 21. Camera pose in relationship to the structure with three cameras at an angular diversity of 30°. Additional cameras were added at equally spaced intervals between the two cameras represented in this figure by cameras 1 and 3. Furthermore, angular diversity was investigated by adjusting the separation between the two end cameras.

For each data set, the location of the nine points were recorded and blurred with a Gaussian function with sigma equal to 0.0001. Empirical tests showed this value of sigma provided consistent results whereas values exceeding 0.001 failed to reconstruct. The blurred points were then supplied to the SfM/MVS reconstruction algorithms for analysis. 100 iterations were performed for each dataset to mitigate variance inherent in any non-linear optimization problem. The reconstruction algorithms implemented followed the techniques described in Chapter II while guidance and coding techniques were adopted from author and professor Jana Kosecka of George Mason University who supplied sample code via her website which accompanies the text, *An Invitation to 3D Vision* [12, 28].

The resulting reconstructions were analyzed for accuracy based on the root mean square distance error (RMSDE) calculated between the original and reconstructed points, x_i and x_r respectively, as seen below

$$\text{RMSDE} = \sqrt{\frac{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2}{n}} \tag{49}$$

where n is the total number of vertices.

3.3 Phase Two: Airborne Collection Profiles

The use of a synthetic 3-D rendering software permited investigation of reconstruction accuracy from images derived by airborne collection platforms without risk to personnel or equipment and allowed for radical flight profiles. This investigation turns to the heart of this effort; determining the optimum flight characteristics required to maximize reconstruction accuracy and completeness. Pursuant to representative intelligence requirements, a specific target within a cluttered scene was selected for target characterization and reconstruction.

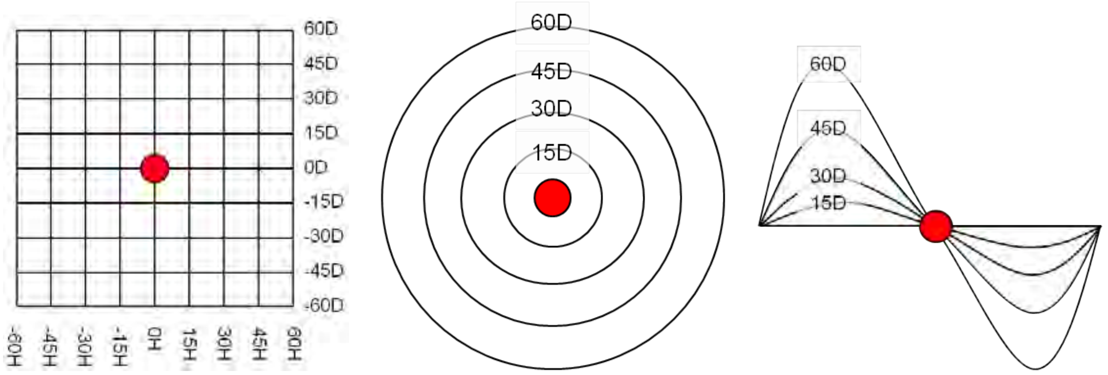
The synthetic scene improves upon the MATLAB simulation due to the realistic nature of the environment by introducing occlusions and requiring the algorithms to locate and associate feature points between images. As noted earlier, Blender [1], an open source 3-D computer graphics rendering package, provided the images required for the reconstruction task. Multiple aerial profiles were collected including linear, circular, and a hybrid s-curve approach. Furthermore, the camera orientation to the aircraft varied from dynamic (agile) mounts possessing the ability to track specific targets to static mounts whose orientation to the airframe is fixed. Details associated to each collection profile and the Blender setup are provided below, but first a brief discussion on the limitations associated with the synthetic environment.

3.3.1 Limitations.

A major limitation to the synthetic environment was the lack of building textures and moving objects such as windblown trees and vehicles. Although Blender allows for both, the necessary texture and motion data files were unavailable at the time of this investigation. Preliminary use of the resulting rendered images revealed the background texture, building edges, and shadowing effects provided sufficient image detail to extract adequate image features and correspondences to produce quality reconstructions. The lack of moving objects within the scene limits the realism; however, their absence only has limited effects on the final reconstruction. As one may note reconstruction vertices are ultimately sourced from scene correspondences, and features derived from moving targets are rejected by feature matching techniques since similar features often cannot be found in subsequent images due to the epipolar constraints used for reconstruction.

3.3.2 Procedures.

The three collection profiles to be examined include linear, circular, and s-curve all of which are visualized in Figure 22. Each pass is identified by the look angle, Θ, measured between the point of closest approach and nadir and ranges from -60° (nadir) to 60° in 15° increments. It is important to understand the difference between the convergence angle described in Figure 3 and the aerial profile look angle seen below.



(a) Linear Flight Profiles (b) Circular Flight Profiles (c) S-Curve Flight Profiles

Figure 22. Visual depiction of aerial flight profiles. In total, three data sets were collected each of the above with a dynamic camera, and a fourth involving a static collection along the linear flight profiles.

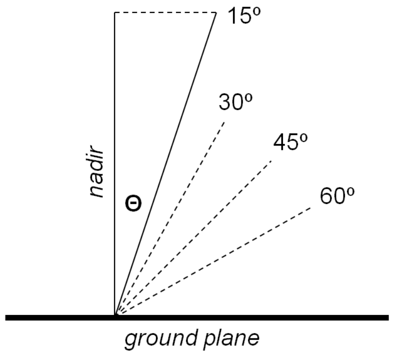
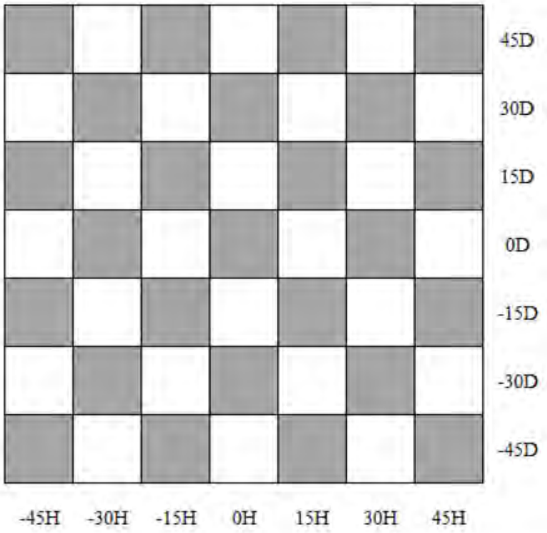


Figure 23. Aerial profile descriptor (look) angles ranging from $-60°$ to $60°$ in $15°$ increments. Negative angles (not pictured) are symmetric.

Linear flight paths are subdivided into either east-west (*D designation*) or north-south (*H designation*) orientation in a cross hatch pattern with the north-south direction indicated by the vertical lines in Figure 22(a). Each pass was completed twice with two different camera properties, dynamic and static. Dynamic camera properties indicate the camera's ability to skew to or track ground targets whereas a static camera's orientation to the aircraft is fixed. In total, twenty eight linear passes were completed encompassing all permutations of pass direction and camera slewing ability. In addition to the single linear passes, multiple linear static passes were combined to determine the level of improvement against the single passes.



-45H	-30H	-15H	0H	15H	30H	45H	
▨		▨		▨		▨	45D
	▨		▨		▨		30D
▨		▨		▨		▨	15D
	▨		▨		▨		0D
▨		▨		▨		▨	-15D
	▨		▨		▨		-30D
▨		▨		▨		▨	-45D

Figure 24. Permutations of `linear_staticcross` passes. The shaded areas represents those linear passes which were combined. For example, along the top row `linear_static_45D1000` was combined with `linear_static_-45H1000`, `linear_static_-15H1000`, `linear_static_15H1000`, and `linear_static_45H1000`. All passes at look angles in excess of ±45° were omitted due to poor reconstruction performance.

The circular flight paths by their nature will always maintain the target in the center of the frame regardless of whether or not the camera is static or dynamic. However for this effort, all circular flight paths will be dynamically controlled. 100 images were acquired for each circular orbit as opposed to the 50 generated from the other passes. This is to ensure complete coverage of target as the dataset can be subsampled at a later date for consistency with other collection profiles.

The s-curve flight paths will be dynamically controlled and follow a sinusoidal pattern passing directly nadir to the target at the closest approach.

Nomenclature for the data sets follows the conventions seen in Table 1. Linear passes transecting the synthetic environment in the east-west direction are designated linear_static(dynamic)_XXD1000 whereas north-south passes follow the nomenclature linear_static(dynamic)_XXH1000. Dynamically controlled and static cameras are differentiated from one another by the second parameter within the filename. Finally, combined passes are referred to as linear_staticcross_XXDXXH and those combined passes limited to only images overlapping the target are referred to as linear_staticcrosslimited_XXDXXH

Table 1. Nomenclature for all flight profiles.

Flight Profile	Nomenclature	Images Used
$Linear(Dynamic)$	linear_dynamic_XXD1000	50
	linear_dynamic_XXH1000	50
$Linear(Static)$	linear_static_XXD1000	50
	linear_static_XXH1000	50
$Linear(StaticCross)$	linear_staticcross_XXDXXH	100
$Linear(StaticCrossLimit)$	linear_staticcrosslimited_XXDXXH	variable
$Circular$	circular_dynamic_XXD1000	100
$S-Curve$	scurve_dynamic_XXD1000	50

3.3.3 Blender.

Blender 2.5 provided the necessary images for the reconstruction effort[1]. The open source software allows users to design and navigate through cityscapes with a variety of lighting, shadowing, and surface textures. As seen in Figure 25, a digital reconstruction of Sadr City, Iraq provided by AFRL/RY served as the urban landscape suitable for this environment. The white buildings reflect the lack of building texture.

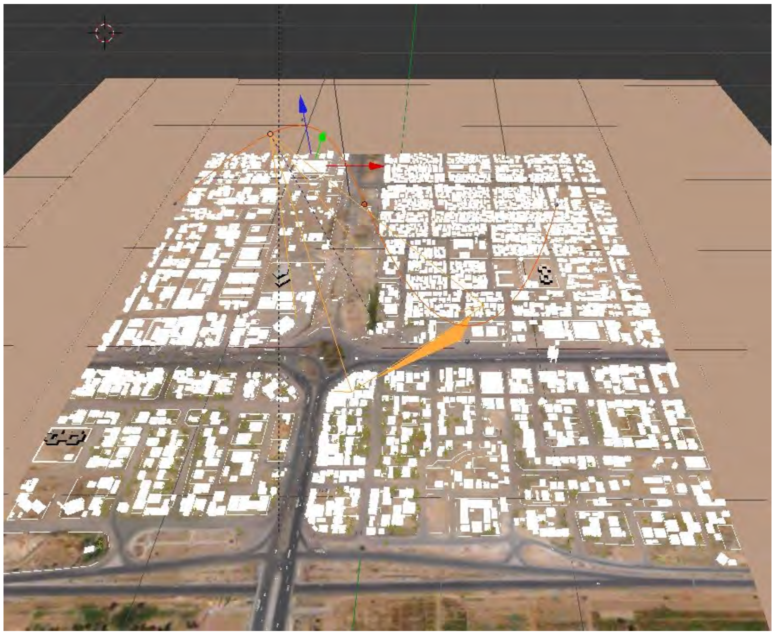


Figure 25. Sadr City synthetic environment.

Various flight paths were constructed within the Blender environment which represented the various aerial profiles noted in the previous section. All flight paths maintained a constant altitude of 1 unit corresponding to a relative altitude of roughly 2000 feet above the target based on the Sadr City scene dimensions and scaling factors. Images were rendered at a resolution of 1000×1000 as these dimensions balanced both the time required for image acquisition and number of features and therefore vertices in the final reconstruction. Furthermore, a camera focal length of 50 mm was used to ensure sufficient structures surrounding the target existed within the image frame. Tests with focal lengths in excess of 75 mm yielded poor reconstructions as the target consumed too much of the individual frames, thereby limiting the number of correspondences since the homogeneous building facets lacked image features. Solar illumination and shadowing was achieved by a uniform irradiance source placed at am angle of 45° to the north resulting in the observed southerly facing shadows. LuxRender, a physical based renderer using unbiased ray tracing techniques simulated the path of individual light rays as they propagated through the scene [7]. Images were rendered via a metropolis unbiased rendering algorithm which terminated when a total of 15 samples were collected by each pixel. A subset of eight images representing the entire 50 image data set of a linear dynamic pass is seen in Figure 26.

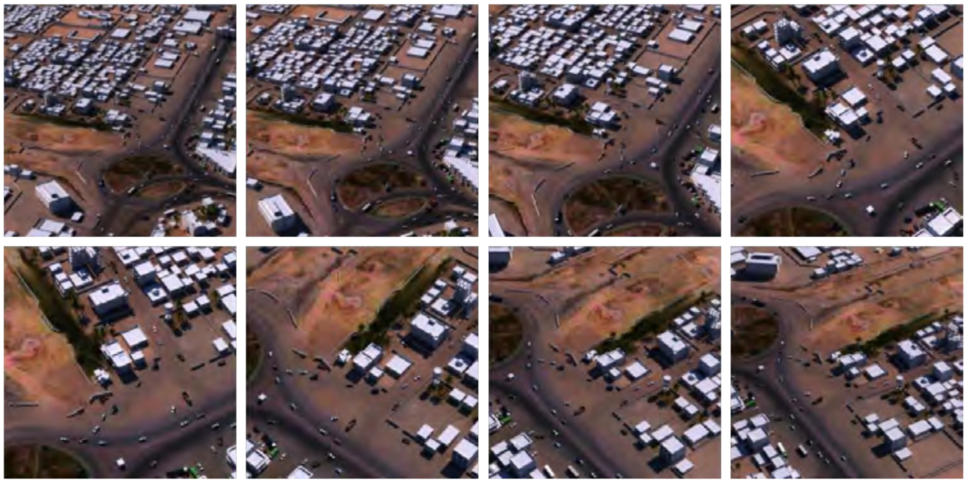


Figure 26. Eight image subset of aerial target collection representing a `linear_dynamic_-15D1000` pass. Centered within each image is the target area consisting of a three building structure.

In Figure 26, the multi-building compound selected for reconstruction is prevalent. The structure exhibits several characteristics quintessential for investigating the reconstruction accuracy and completeness based on flight profiles. First, its location within the center of the city allows for low viewing angles while urban terrain and buildings still remain in each image. Selection of an edge target would severely limit the number of reconstruction vertices and only those features around the target would be selected for reconstruction. With numerous structures, and therefore features within each image, the algorithm must effectively manage clutter surrounding the intended target, and those correspondences originating from scene clutter allow for additional refinement of the fundamental matrix and minimization of the non-linear bundle adjustment solutions. Second, and in seemingly contradiction to the first characteristic, the compound is relatively unobstructed when viewed from the west and south while possible occlusions exist when viewed from the opposing directions. Although the compound is unobstructed from the west and south, the ground surface and road will provide distinct features and correspondences. Finally, all buildings within the compound and surrounding area represent a variety of equally distributed heights normalized to the tallest structure. This final characteristic is of immense importance as multiple building heights provides sufficient basis to judge the algorithms accuracy to extract depth information.

3.4 Data Processing

Multiple software tools exist which offer an automated process to extract features, relate images, and provide a 3-D reconstruction. For this effort the software package Bundler in conjunction with PMVS2 and CMVS were used. All software packages provide the necessary binary and executable files necessary for correct installation which are available for download from the University of Washington's Computer Science and Engineering Photo Tourism website [2, 11, 5].

The data workflow from scene generation to dense point cloud is seen in Figure 27 to provide guidance for the remainder of this section.

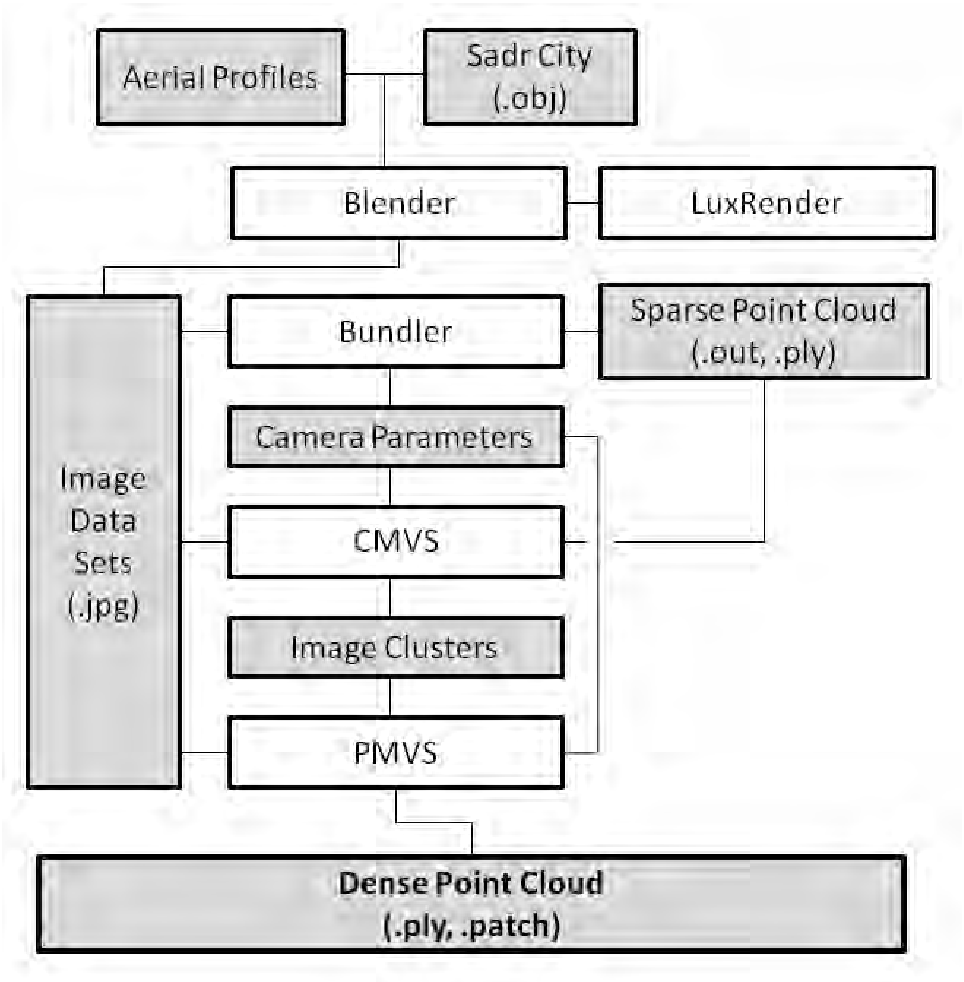


Figure 27. Basic data workflow constituting the major dense reconstruction steps. Shaded cells represent data inputs and products.

3.4.1 Bundler.

All Blender derived images were processed with the SfM techniques discussed in Chapter II using the Perl based software package, Bundler. Bundler, developed by Noah Snavely, serves as the core backbone to Microsoft's Photosynth image processing platform and determines the basic sparse structure and camera parameters from unordered image sequences [2, 9]. Bundler consists of four core subfunctions: extraction of focal length from image metadata, image feature extraction using SIFT, feature matching using Approximate Nearest Neighbor, and finally, application of the SfM algorithm with point and camera refinement through the minimization technique, Sparse Bundle Adjustment.

Extraction of the image focal length aids immeasurably in initializing the fundamental matrix by providing a partially calibrated camera view. It has been shown calibrated cameras produce Euclidean reconstructions when both the rotation and translation between cameras are known or derived from the SfM process [23, 28]. Therefore seeding the fundamental matrix calculation with partially calibrated images greatly enhances the solution's accuracy. Once extracted, the optical focal length must be converted from metric units, typically mm, to pixels by

$$\text{focal length}_{\text{pixels}} = \text{image width}_{\text{pixels}} \times \left(\frac{\text{focal length}_{\text{mm}}}{\text{sensor width}_{\text{mm}}}\right) \tag{50}$$

where image and sensor widths are measured along the major axis of the image, and the focal length measured in either pixels or millimeters as noted. For example, base images for the AFIT block and AFIT campus reconstruction examples were acquired with a 50 *mm* lens mounted on a Canon 40D sensor body. The Canon 40D contains an APS-C CMOS sensor array measuring 22.2×14.8 *mm* capable of recording images with a maximum resolution of 3888×2592 pixels. Under these conditions, the corresponding focal length is 8756.76 pixels. Note the individual APS-C CMOS sensors are square, and therefore, usage of the width or height produce identical pixel focal lengths. All images acquired through Blender maintain a focal pixel length equal to 1000 corresponding to the dimensions of the image. For the remainder of this effort, all focal length measurements will be conveyed in pixel units not metric units which typically define optical systems unless explicitly stated otherwise. This distinction allows for the inclusion of sensor characteristics which reflect the actual conversion from the real world to digital coordinates and are required for true Euclidean reconstructions.

Secondly, the images are related to one another through extraction of feature points via the SIFT algorithm. However, Bundler only uses SIFT to extract the keypoint descriptors from each image and relies on the Approximate Nearest Neighbor (ANN) to register image features. ANN, developed by David Mount, constructs a kd-tree from the keypoint descriptors and determines nearest neighbors based on the euclidean distance [31]. The program uses the SIFT generated keypoint descriptors from the first image and recursively navigates through the kd-tree minimizing the Euclidean distance for a keypoint description from a second image. A final survey of data points in neighboring cells verifies the selected point as a possible match since closer points in neighboring cells may exist. The resulting data point with the minimum Euclidean distance are matched and designed as a matched keypoint.

The final function performed by Bundler is application of the SfM algorithm. In this step, Bundler extracts matched keypoints and iteratively solves for the fundamental matrix using RANSAC to derive the relative camera pose assuming the first camera is positioned at the origin. Once a solution for the camera parameters has been determined, the 3-D points are triangulated between sequential images. Both camera parameters and sparse vertices are exported in the `bundle.out` file.

3.4.2 CMVS/PMVS2.

The CMVS/PMVS2 software packages were used in conjunction with Bundler to derive a dense point cloud [5, 11]. As stated in Subsection 2.7.2, CMVS conditions the image sets by removing redundant images in which the small baseline between cameras degrades reconstruction accuracy and clusters neighboring images for use on multi-core systems. The image clusters produced by CMVS with input from GenOption, a program initializing CMVS parameters such as those mentioned in Table 2, are passed to PMVS through a series of `option-####.txt` files. These files contain the necessary clustering and PMVS commands for the dense point cloud reconstruction. Depending on the clustering output of CMVS, multiple `option-000(...).txt.ply` files may exist. These additional files serve as processing instructions suitable for parallel computing on multicore computers. Furthermore, CMVS creates a `vis.dat` file which contains the images which PMVS uses as well as camera contour files containing the necessary pointing and location information for each camera. The contour files are suitably named, `####.txt` where the `#` signs represent the camera number $(0 - n)$, and are derived from the `bundle.out` file. With these files, PMVS reconstructs a dense point cloud following the same methodology as described in Subsection 2.7.1 by constraining the search for additional correspondences within two pixels of the epipolar lines.

CMVS/PMVS2 allows for the selection of several thresholds and parameters when reconstructing the dense point cloud. The variable, value used, and meaning are found in Table 2. Preliminary testing showed processing time increased by a factor of four when the images were retained at full size with $level = 0$, whereas $level = 1$ reconstructed a similar number of points without a significant loss of data points.

Table 2. Selection parameters used for CMVS/PMVS2. All reconstructions used the same values. A final parameter denoting the number of available processors was hard coded to 8.

Variable	Value Used	Nomenclature
clustersize	30	Denotes the maximum number of images per cluster. This variable is largely depended on the available computer memory.
level	1	$level = 0$ denotes the image is used with full resolution whereas an incremental increase in the *level* effectively halves the images. For instance, $level = 1$ images are halved to one quarter of the original pixels and $level = 2$ only a sixteenth of the images are used.
csize	2	Controls the reconstruction density by attempting to reconstruct one path in every $csize \times csize$ pixel region in the target images.
threshold	0.9	Photometric consistency measure. Only patches passing this threshold will be retained.

3.5 Data Analysis

It is imperative to quantify reconstruction accuracy which is defined as the level to which the reconstruction resembles the original target. Several measures will be used including root mean square distance error between selected data reference points and completeness or the degree to which all surfaces were reconstructed.

3.5.1 Root Mean Square Distance Error.

Since the Sadr City synthetic environment provides the structure geometry, the exact dimensions and relationships between the various structures is known. The `sadrcity.obj` file provides the vertex and facet information required for each building. As seen in Figure 28, twenty six specific vertices were recorded and stored as reference points. These points serve as the ground truth by which the reconstruction will be measured. Four buildings within the target area were selected for RMSDE analysis. The selection of these particular buildings fulfilled several functions. First, the distribution of the vertices in relationship to the surrounding structure allows for occlusion of several vertices at oblique viewing directions especially those between the buildings. Secondly, the height distribution of the vertices is equally spaced within the z-direction which provides multiple opportunities to determine the SfM/MVS algorithm's ability to exact accurate depth.

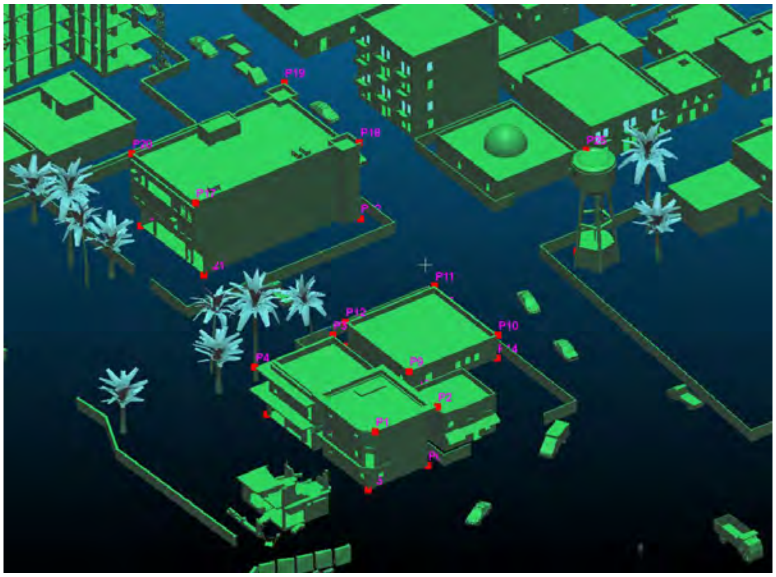


Figure 28. Twenty six identified reference markers within ground truth.

Specific vertices within the point cloud reconstructions representing these twenty six points were selected based on their proximity to the real point. Since particular viewing geometries never image one side of the building it is logical to assume no vertex will be reconstructed at that point. Therefore, the closest reconstructed point must be selected to best represent the ground truth vertex. Once all twenty six reconstructed vertices are extracted, the RMSDE between the points was recorded.

3.5.2 Localized Point Density.

To measure the completeness of the reconstruction, the localized point density of the vertices within the target areas was determined. Completeness is measured by the number of vertices within a cube encompassing the entire target area. This measure provides several sources of information. First, the number of vertices within the target area directly relates to the RMSDE as additional vertices improve the selection of the correct vertex representing one of the reference points. Secondly as the number of vertices increase, a corresponding increase in structure detail is observed, further aiding in the selection of the comparison points and representation to the original structure.

3.6 Data Processing Computer

The processes described above require immense computational capabilities, and reconstruction results are highly dependent on the specific computer. For instance when the reconstruction workflow was performed on identical datasets, the number of reconstructed points differentiated by roughly 10% depending on the computer and in particular the amount of available RAM. Recognizing this ambiguity between computers, a single computer was used to perform all reconstructions. The computer used employed two quad core Intel® Xenon® X5667 CPUs operating at 3.07 Mhz each supported by 12.0 GB of available RAM.

3.7 Chapter Summary

Quantifying the effects of viewing geometry on the subsequent 3-D reconstruction accuracy is dependent of numerous parameters and requires a multi-phase effort. In the first phase, a MATLAB algorithm with the ability to reconstruct a simple geometric shape provided the basic analytical framework to support in-depth study into the effects various reconstruction parameters on the final reconstruction. By providing the user full control of the camera pose, calibration, number of camera frames supplied to the algorithm, geometric shape to be constructed, and finally a qualitative comparison to the original structure, the investigator was able to interrogate the algorithm and environmental effects on the reconstruction with ease. This arrangement offered unparalleled access to the reconstruction process and provided the necessary insight into the effects of convergence angles and number of frames on the final reconstruction. Within the second phase, a synthetic 3-D environment was used to generate images simulating a variety of aerial collection profiles similar to typical reconnaissance flight profiles such as linear one pass flybys and circular orbits each equipped with static and agile cameras. In addition, a hybrid s-curve profile exhibiting characteristics of both the linear and circular profiles was explored to fully determine the optimum collection geometry to support operational scenarios. This secondary effort supports the first by introducing real world effects such as variable illumination loading, obscured points, and bridges the gap between laboratory algorithms and real-world operations.

IV. Results

4.1 Chapter Overview

This chapter covers the results of both the MATLAB simulation code as well as the Blender and CMVS/PMVS2 derived point cloud reconstructions. The MATLAB simulation code successfully reconstructed the nine point structure under a variety of conditions including limited camera frames and convergence angles. Reconstructions from the Sadr City synthetic images were much more diverse; whereas most reconstructions clearly represented the scene, several produced erroneous results.

4.2 Simulation

Projective reconstruction of the nine point structure was successfully achieved for all convergence angles, 1° to 100° in 1° increments, and number of cameras which ranged from 2 to 20. Without foreknowledge of the camera calibration matrix, K, or a-priori scene knowledge, the results were limited to a projective reconstruction as seen in Figure 29. As expected the general structure of the object was retained however significant distortion exists.

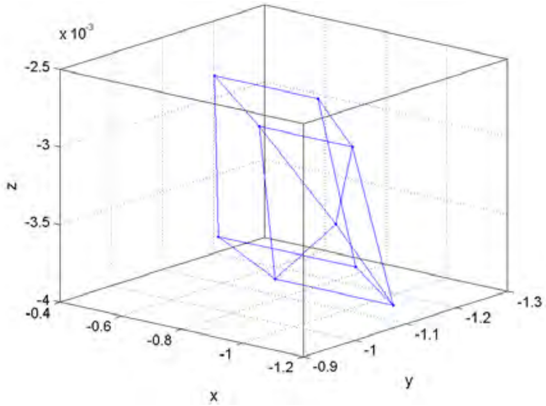


Figure 29. Projective Reconstruction.

In Figure 30, three vanishing points were computed using the intersection points between the vectors, $\overrightarrow{12}$ and $\overrightarrow{56}$, $\overrightarrow{14}$ and $\overrightarrow{23}$, and finally $\overrightarrow{15}$ and $\overrightarrow{26}$. The affine upgrade correctly projects the image preserving the basic geometric shape. However, the reconstruction is still limited to a non-Euclidean reconstruction exemplified by the rectangular cuboid shape as opposed to the true cubic structure.

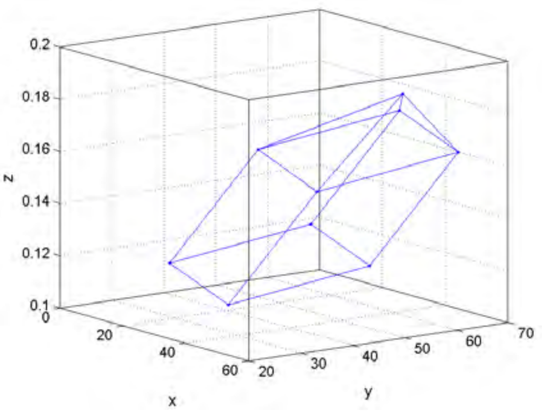


Figure 30. Affine Reconstruction.

The final Euclidean reconstruction requires the camera calibration matrix to normalize the projection matrices, $\Pi_{1e} = [K, 0]$ and $\Pi_{2e} = [KR, KT]$. The reconstruction was initialized by computing the fundamental matrix from the two extreme frames. Next the remaining images are iteratively added using a rank based factorization to complete the multi-view reconstruction seen in Figure 29. As opposed to using a stratified reconstruction process where the Euclidean reconstruction would be created in a projective, affine, Euclidean sequence, a direct perspective to Euclidean upgrade was employed. To complete this upgrade, the camera calibration matrix must be determined directly from the images provided. K was determined by initially guessing the focal length of the camera, assuming a pixel skew equivalent to zero, and calculating the center of the image as the geometric center of the image.

These values were then used as the baseline to refine the focal length using absolute quadric constraints. Once the focal length was determined, the projective structure was upgraded to Euclidean as seen in Figure 31.

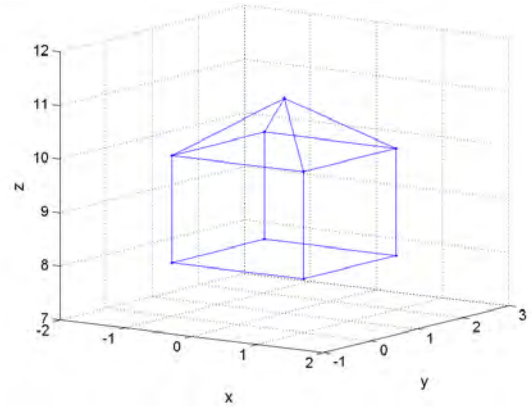
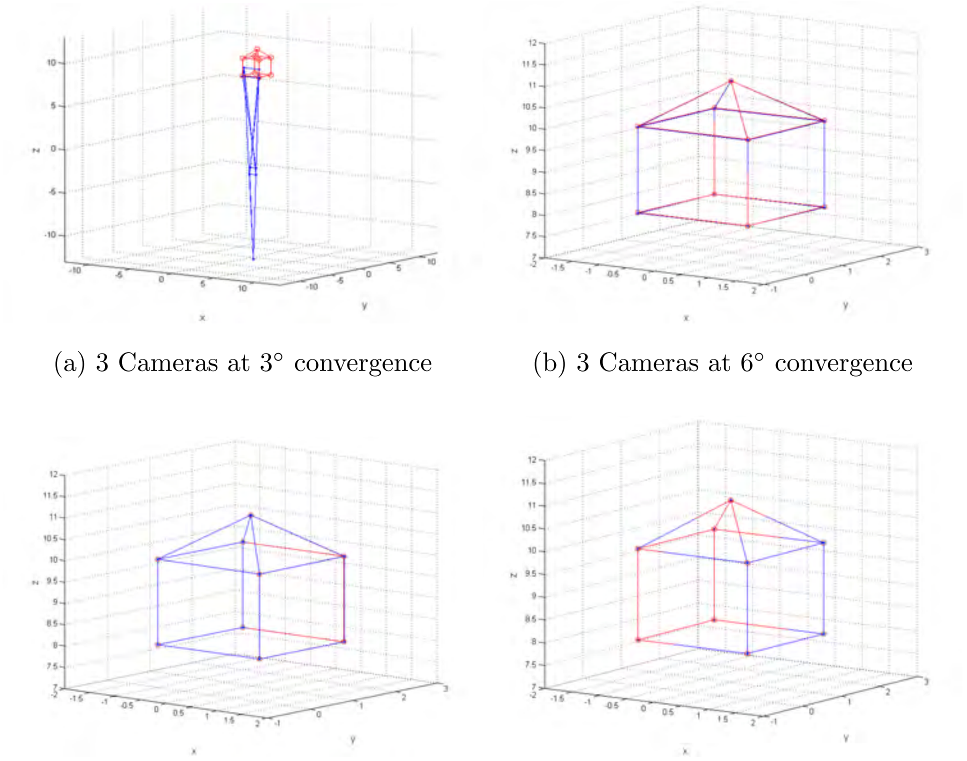


Figure 31. Euclidean Reconstruction.

The Euclidean reconstruction seen in the above figure was reconstructed using five camera frames and a convergence angle of 30°. The reconstruction has a RMSDE on the order of 3×10^{-14} thereby providing a realistic Euclidean representation of the true structure.

The above process was repeated at all convergence angles from 1° to 100° in 1° increments while varying the number of cameras from 2 to 20. When only 2 cameras were used, the reconstruction algorithm frequently failed due to inconsistent computational results. Several reconstructions at varying test parameters are shown below.



(a) 3 Cameras at 3° convergence

(b) 3 Cameras at 6° convergence

(c) 7 Cameras at 30° convergence

(d) 20 Cameras at 100° convergence

Figure 32. Reconstruction of nine point structure at a variety of parameters. Note the dramatic improvement at angles greater than 5°.

These reconstructions highlight several peculiarities with this specific reconstruction algorithm. First, note the substantial improvement in accuracy at convergence angles exceeding 5°. Furthermore, the apparent accuracy improved only slightly despite the additional camera frames and wider convergence angles. Finally, with an initial focal length estimate of 400, the focal length was resolved to 497.8800, 498.4969, 500.0067, and 500.0291 in each of the four scenarios above while the actual focal length was 500.

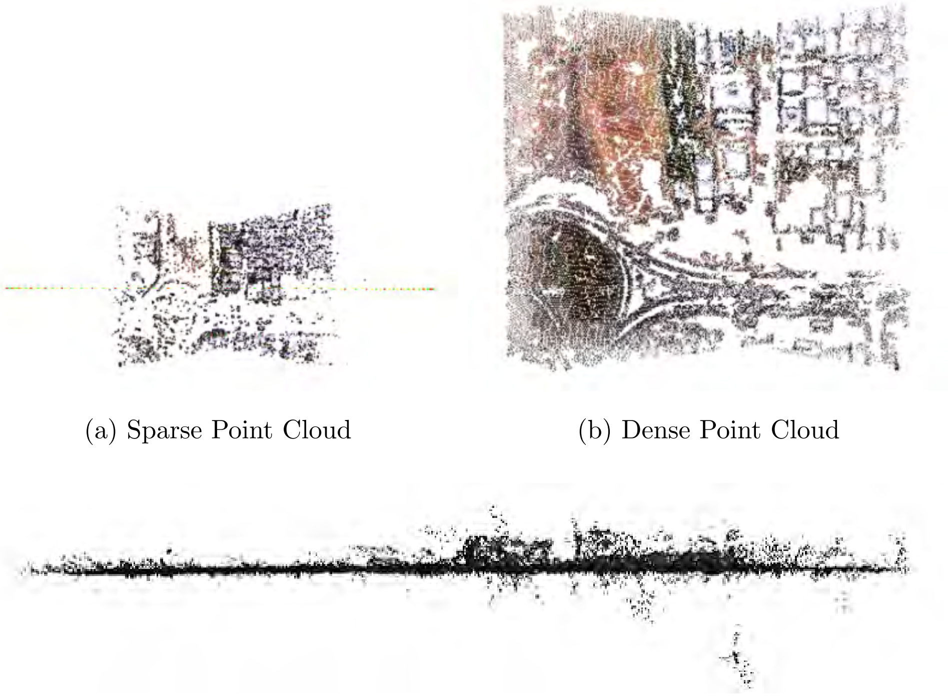
4.3 Aerial Collections

The Blender derived image sets represent each of the various flight profiles resulted in 44 unique collection perspectives and 2500 images. The intent of this chapter is to demonstrate the SfM/MVS workflow's ability to reconstruction a complex urban scene at a variety of flight profiles detailed in the previous chapter. Therefore discussion to each is limited to only immediate observations, whereas detailed analysis concerning each reconstruction occurs in the following chapter. Example point clouds from the six flight profile categories are provided, but all may be found in the Appendix. For each flight profile both the sparse and dense point clouds are depicted at varying perspectives to orient the reader to each reconstruction. In the upper left corner, a sparse point cloud is included which also contains the SfM determined camera centers which are denoted by the red and green alternating pixels. The yellow pixels show the pose for each camera center. The upper right corner contains the dense point cloud reconstruction from a top-down perspective to observe the x-y spatial reconstruction. Finally, the dense point cloud is again seen as the final perspective which highlights the vertical reconstruction along the z-axis.

In general, the results varied from extraordinarily dense and representative of the target area to failed reconstructions where the scene structure is unidentifiable. Those profiles possessing both a variety in target magnification and diverse viewing angles provided rich datasets particularly suited to well behaved reconstructions.

4.3.1 Linear Flight Profile - Dynamic, `linear_dynamic_XXX`.

In total, 18 flight profiles were recorded at a variety of look angles from −60° to 60° in 15° increments. Initial inspection revealed the profile's ability to provide a robust representation of the target area independent of viewing geometry from nadir extending to 45°. The combination of angular diversity and variable target magnification supplied the reconstruction algorithm with a wide variety of data points Only `linear_dynamic_60D1000` failed to provide a reconstruction characteristic of the original target area.



(a) Sparse Point Cloud

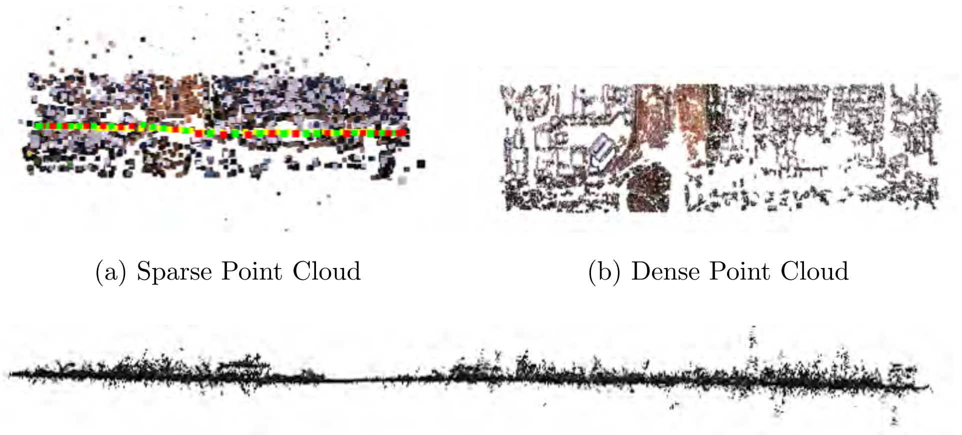
(b) Dense Point Cloud

(c) Dense Point Cloud - Ground Plane

Figure 33. linear_dynamic_0D1000. The yellow, red, and green pixels contained within the sparse point cloud (a) represent the camera centers as determined by Bundler.

4.3.2 Linear Flight Profile - Static, `linear_static_XXX`.

Linear static flight profiles provided a stark contrast to the dynamic reconstructions seen in Figure 33. Of the 18 total flight profiles, only 13 provided recognizable reconstructions with all poor reconstructions occurring at nadir or near nadir angles. The relative homogeneous data sets lacked the angular and range diversity required to compute the fundamental matrix.



(a) Sparse Point Cloud

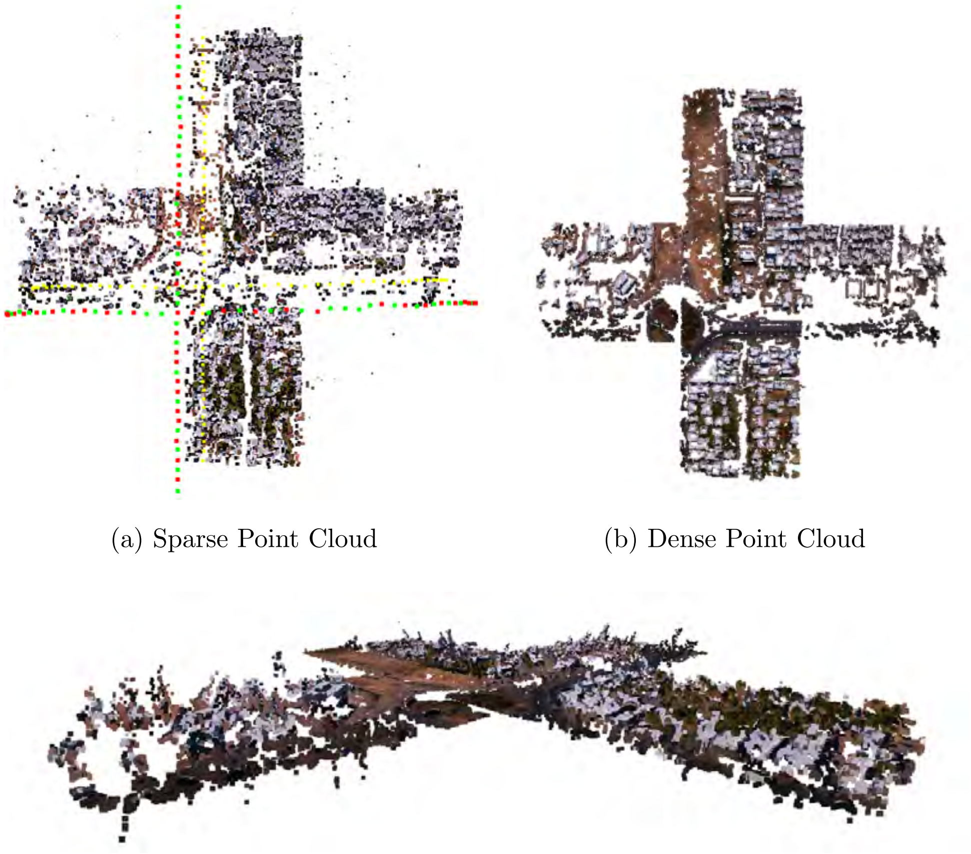
(b) Dense Point Cloud

(c) Dense Point Cloud - Ground Plane

Figure 34. linear_static_-15D1000. The yellow, red, and green pixels contained within the sparse point cloud (a) represent the camera centers as determined by Bundler.

4.3.3 Linear Flight Profile - Multiple Profiles, `linear_staticcross_XXDXXH`.

When the linear static profiles were combined with orthogonal passes, a significant improvement was observed in reconstruction accuracy. However, 6 of the total 25 combined linear static profiles failed, roughly the same percentage as the linear static profiles. Nevertheless, successful reconstructions exhibited improvement in accuracy due to the increased viewing angles covering additional areas of the target.



(a) Sparse Point Cloud

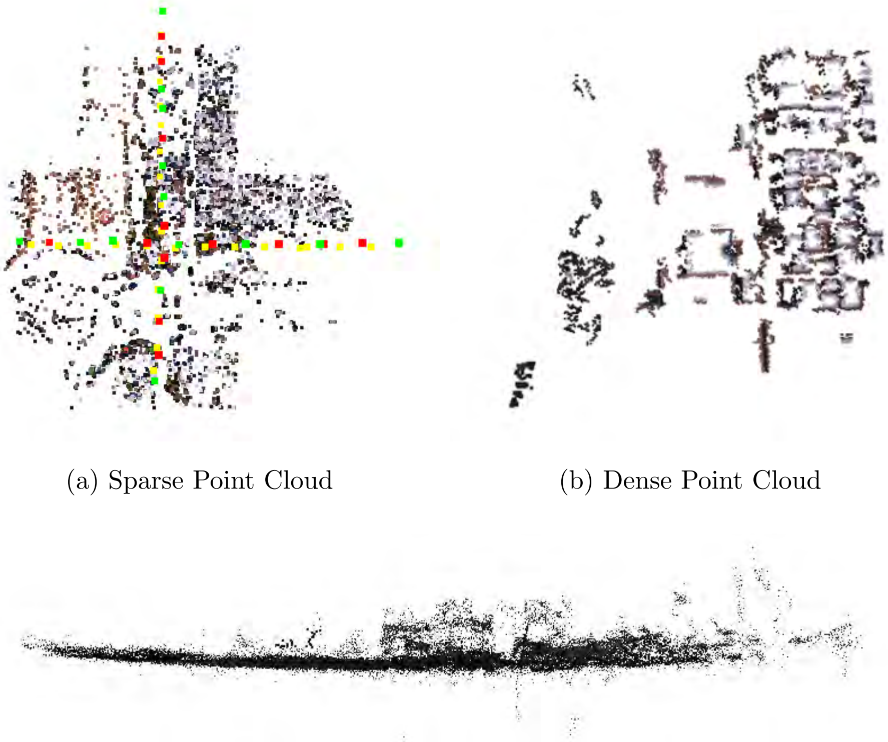
(b) Dense Point Cloud

(c) Dense Point Cloud - Ground Plane

Figure 35. linear_staticcross_-15D-15H. Note the reduction in vertical error occurring within the regions of overlap.

4.3.4 Linear Flight Profile - Multiple Profiles Limited, `linear_staticcrosslimited_XXDXXH`.

Limiting the supplied images to only those containing the target scene and overlapping images from the orthogonal pass produced incredibly realistic results. In several instances, this method of data conditioning provided the only means by which successful reconstructions were finally achieved as seen by the pure nadir collections reconstructions of the target. Previous independent and fully combined passes failed to reconstruct these profiles.



(a) Sparse Point Cloud

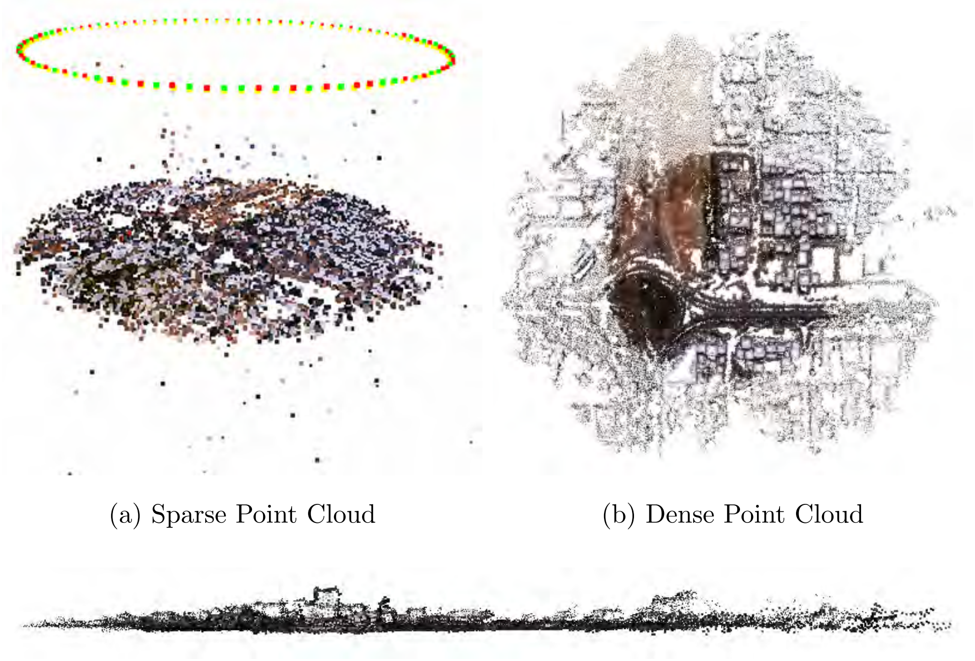
(b) Dense Point Cloud

(c) Dense Point Cloud - Ground Plane

Figure 36. linear_staticcrosslimited_0D0H. Although significantly better reconstructions exist to depict this data set's ability to reconstruct the scene (see appendix), this example presents a unique case. In three other situations where the passes were used individually or combined in their entirety failed. By limiting the applied images to only those with similar imaged regions, the reconstruction succeeded.

4.3.5 Circular Flight Profile, `circular_dynamic_XXX`.

The reconstructions observed from the circular profiles provided high quality representations of the target area with exception of the 60° profile. By viewing the target area from all directions in a complete 360° circle, the profile maximizes the chances of observing and recording all facets of the target area. Despite the constant target range, the extensive angular diversity provided sufficient information into the solution space to determine a robust fundamental matrix relating the images.



(a) Sparse Point Cloud

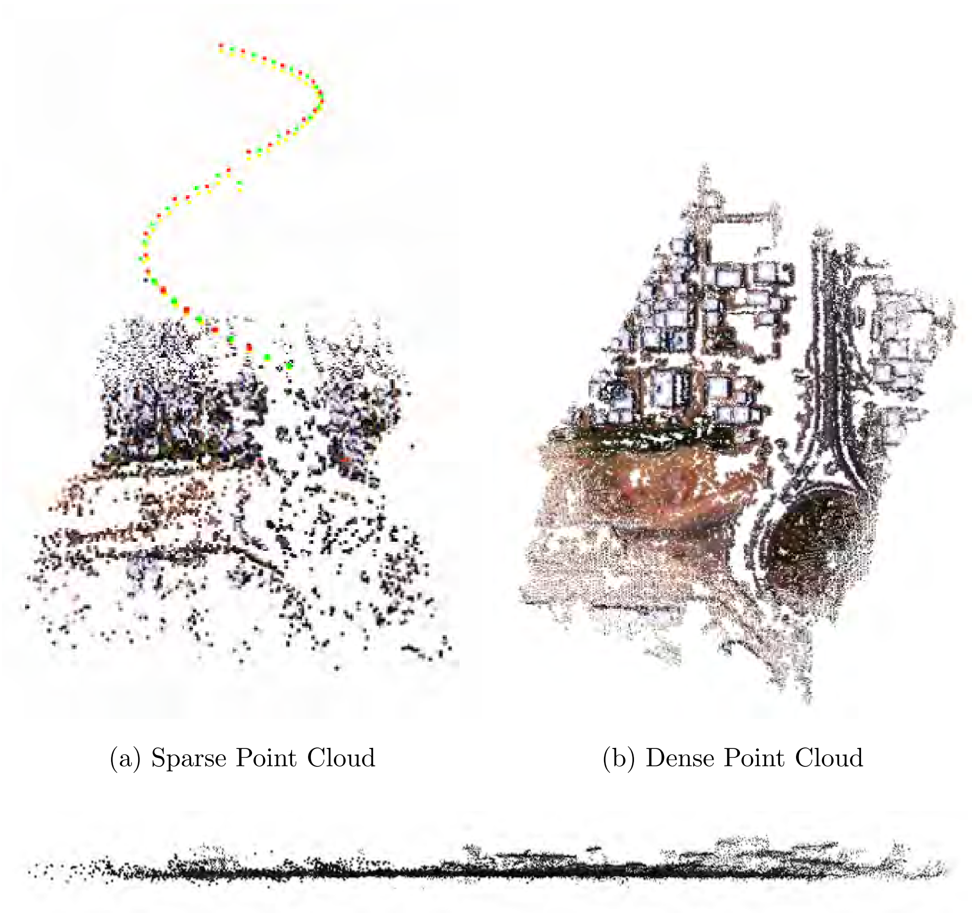
(b) Dense Point Cloud

(c) Dense Point Cloud - Ground Plane

Figure 37. circular_dynamic_45D1000. Dense point clouds originating from the circular flight profiles exemplify the ability of PMVS2 to find additional correspondences within regions of overlap. The higher density of points contained within the reconstruction's center is prevelant.

4.3.6 S-Curve Flight Profile, `scurve_dynamic_XXX`.

Of all the flight profiles, those constructed from the s-curve flight profile provided the most accurate reconstructions due to the combination of angular diversity and target magnification. All reconstructions were successful with exception of the 60° profile.



(a) Sparse Point Cloud

(b) Dense Point Cloud

(c) Dense Point Cloud - Ground Plane

Figure 38. scurve_dynamic_30D1000. Note at discontinuity in camera center positioning occurring at the nadir point. This feature grows until complete reconstruction failure at 60° when the discontinuity's severity precludes determination of camera centers on both sides of nadir.

4.4 Chapter Summary

Both simulated MATLAB reconstructions of a simple geometric structure and those of a synthetic 3-D environment using academic software packages were accomplished. The MATLAB reconstruction of a nine point structure allowed variance in the convergence and number of cameras to investigate the dependency of reconstruction accuracy on these parameters. In total 190,000 individual tests were completed covering all permutations by varying the convergence angle from 1° to 100° in 1° increments while the number of camera frames at each convergence angles from 2 to 20. In regards to the synthetic 3-D environment, 44 collections encompassing linear, circular, and s-curve flight profiles with a mixture of static and dynamic cameras totaling 2500 images. 7 of the 44 profiles failed to provide sufficient reconstructions enabling identification of the target area with all originating from the linear static passes or those in excess of ±45°.

V. Analysis

5.1 Chapter Overview

The results seen in the previous chapter verify the SfM/MVS algorithm's ability to reconstruct a 3-D point cloud from 2-D images. Within the MATLAB simulation, it was noted variances in the number of camera frames and convergence angle produce significant effects on reconstruction accuracy. This chapter seeks to add fidelity and quantify this relationship to determine the relationship between these parameters on reconstruction accuracy. Furthermore, the Sadr City dense point cloud results will undergo an extensive analysis where the effects of viewing geometry on the accuracy and completeness of the reconstruction will be investigated. It will be shown rich datasets which contain significant angular and spatial diversity provide the best data to accurately reconstruct the scene. Finally, rationale why specific image sets achieve superior results is investigated by probing the image distribution within the algorithm in which an interesting multi-mode structure is observed.

5.2 Effects of Convergence Angle and Number of Cameras

The MATLAB simulation effort produces a variety of object reconstructions representative of the diverse datasets. While constraining the number of camera frames and convergence angle, the accuracy of the reconstruction decreases to nearly unrecognizable results when the convergence angles was less than 6° or when less than 3 cameras were used. This provides a baseline for the minimum requirements necessary for an accurate reconstruction. Furthermore once these thresholds are achieved, RMSDE continues to decrease as either the number of cameras or convergence angle increases as seen Figure 39 denoting a interdependence on both the number of camera frames and their spatial distribution.

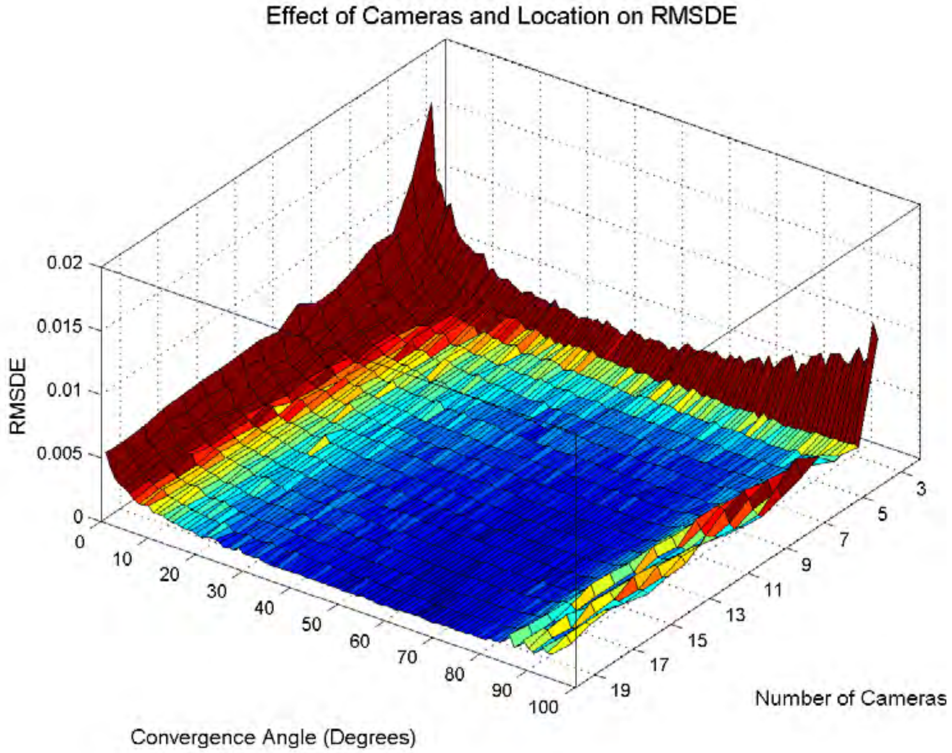
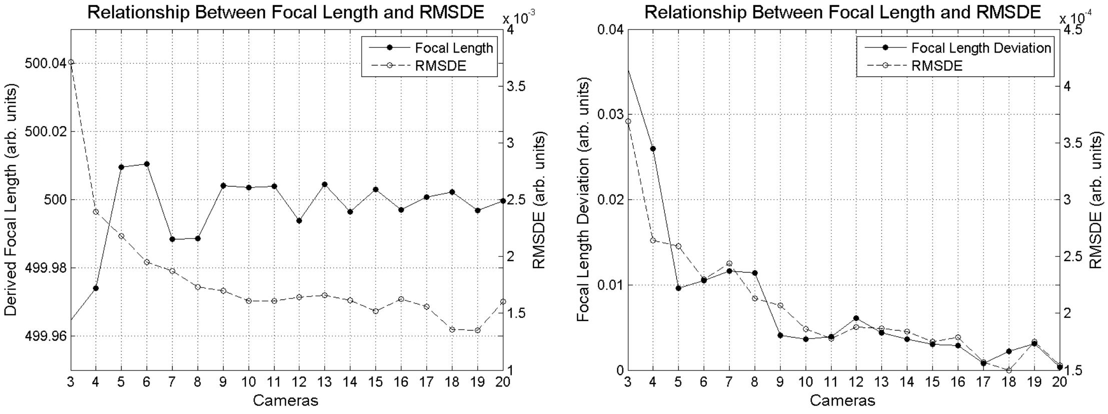


Figure 39. Effect of convergence and number of frames on reconstruction accuracy of a simple nine point structure. Note the large minimum highlighting the reconstruction algorithm's robustness to variable collection geometries. Operationally this provides potential users a wide operational envelope in which imagery may be collected and still obtain an accurate 3-D scene reconstruction.

The number of camera frames directly effects the resulting accuracy by controlling the amount of information supplied to the SfM/MVS algorithms. Therefore it is expected additional cameras would improve the scene reconstruction accuracy as the above figure indicates, but physical explanations why the algorithm fails with two cameras and subsequently improves with the inclusions of additional cameras remains to be determined. Both of these issues are addressed below. Two camera reconstructions failed because the the autocalibration process contains insufficient information to correctly estimate the camera focal length. When the absolute quadric constraint appears in the form found in Equation 45 the five unknowns, $[a_1, a_1, a_1, a_1, a_1]$, may be recovered with the four constraints seen in Equation 46 for each image pair [28]. In such case three views are required for a unique solution. When less than three cameras are employed, the focal length estimation often results in imaginary or grossly underestimated real focal lengths centered about zero. Regardless, when this erroneous focal length is inserted into the camera calibration matrix, K, and applied to the correspondences to achieve a partially calibrated view, the projective reconstruction contains significant errors. Furthermore, the partially calibrated data points serve as the foundation for the rotation and translation estimates during the rigid body approximations to extract the projective to euclidean upgrade matrix. Ultimately, the resulting rotation and translation matrices derived off the erroneous partially calibrated views results in null or imaginary values.

This problem is alleviated by increasing the number of camera frames to correctly estimate the focal length. At three camera frames, the focal length with an initial value of 400 is correctly estimated to a value of 499.94 of the true 500.00 with unitless dimensions. Additional camera frames continue to increase reconstruction accuracy eventually reaching on observed focal length estimate of 500.00 ± 0.001 at twenty cameras. Once this threshold is met, the additional information allows for solution refinement which steadily improves the estimated focal length and therefore reconstruction accuracy. Figure 40 accurately illustrates this relationship.



(a) Derived focal length

(b) Deviation from actual focal length

Figure 40. Relationship between focal length and RMSDE illustrating the effects of improved focal length estimation on the resulting RMSDE accuracy.

The left image depicts the gradual refinement of the focal length as the number of camera frames increase alongside the resulting RMSDE improvement. The right image clarifies this relationship by showing the proportional effect of decreasing absolute focal length error and accuracy. Subsequent analysis determined the effects of the initial value had little effect on the algorithms ability to rapidly converge to the accurate focal length. When seeded with initial values ranging from 10 to 10000, the algorithm converged to 500 ± 2 within the first three camera frames and steadily improved with additional camera frames. As an extreme case representing the true robustness of the algorithm, an initial seed value of 1×10^6 required only six camera frames to converge to a value of 500 ± 2. It is also noted additional camera frames beyond the initial three contribute little to the overall accuracy of the reconstruction as seen in the scale of the right axis. Therefore unless absolute accuracy is required three camera frames will suffice to generate an accurate reconstruction with minimal computational demand.

As the convergence angle increases, a relationship similar to that seen with increasing the number of camera frames is observed. With convergence angles less than 6°, the reconstruction poorly represents the scene as seen in Figure 32a with RMSDE exceeding 8.0 where typical reconstruction have RMSDE on the order of 10^{-4}. The reconstructions were so poor, Figure 39 does not include RMSDE associated with convergence angles less than 6° since the extreme RMSDE range would eliminate the resolution at lower ranges. Whereas the camera threshold was dependent on the extraction of a correct focal length, the convergence angle threshold is related to the accurate determination of the R and T matrices which define the motion between cameras. When the cameras are only separated by a marginal baseline at convergence angles less than 6°, the algorithm struggles to determine the exact placement of the cameras in relation to one another. The poor placement of the camera centers result in projective reconstructions with negative z-depths, which in return, frustrate the Euclidean upgrade once an accurate focal length is estimated.

Increasing convergence angles beyond the minimum 6° improves the reconstruction accuracy in much the same way increasing the number of cameras provided refinement of the focal length. This improvement directly correlates to the improved camera pose determination. Within the MATLAB environment the exact y-translation between camera frames is controlled during the acquisition of each image. These images then form the basis of the final Euclidean reconstruction which requires determination of the relative K, R, and T components. The K matrix has already been discussed in the preceding analysis and is well formed at 10 cameras which serves as the basis for the remainder of this section. To illustrate the effects of error within the R and T matrices on the reconstruction, a number of simulations were run with ten cameras and increasing convergence angles. Since the camera was translating along the y-axis in incremental steps, the translation of the camera derived during the reconstruction should match that of the initial input to a scale factor. Figure 41 illustrates the effect on RMSDE compared to the absolute error in the y-component of the translation vector identified in Equation 6. 300 iterations were averaged to achieve a statistically relevant number of data points.

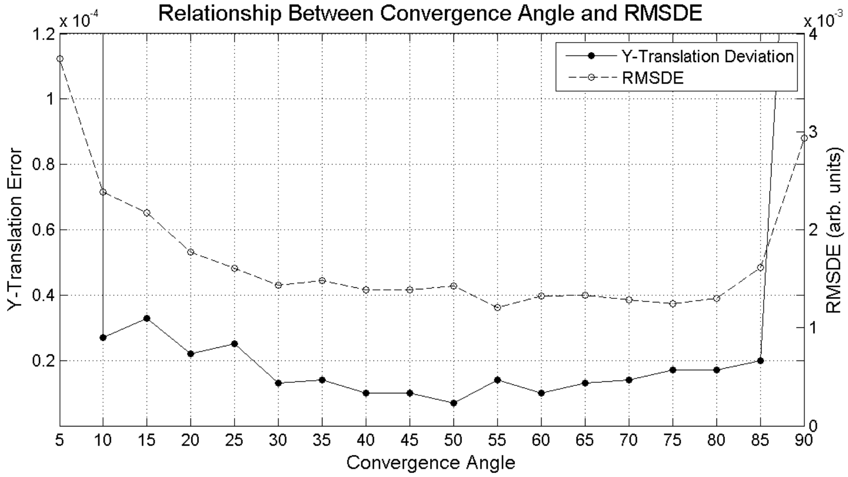


Figure 41. Relationship between translation error and RMSDE.

As the y-translation error increases, accuracy is negatively effected. At small convergence angles the algorithm struggles to extract the rigid body motion of the camera to determine the proper translation vector. However as convergence angle increases, translation determination improves. As a point of order, the rotation matrices were also analyzed, and the derived values closely matched those used to acquire the images. Thus, they were not addressed further.

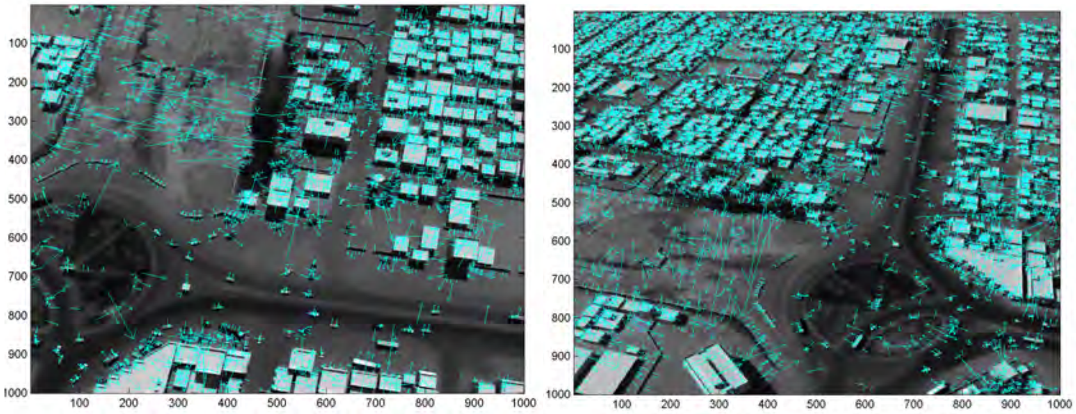
In summary, the number of cameras and convergence angle have a profound effect on the ultimate reconstruction accuracy. Thresholds exist for each parameter as the number of camera frames must be at least three and the convergence angle must exceed 5°. These two thresholds are interrelated through the camera projection matrices, $\Pi_{1e} = [K|0]$ and $\Pi_{2e} = [KR, KT]$, seen at the conclusion of Chapter II. Both Π_{1e} and Π_{2e} contribute to the final 3-D reconstruction and are dependent on accurate values for K, R, and T. Unfortunately when only 2 cameras are used, K is erroneous, and when the convergence angle is less than 6°, R and T exhibit anomalous values. Finally, additional cameras or wider convergence angles improve the reconstruction accuracy, but computational time and resources are adversely affected for little improvement in this situation when all scene vertices are known. With real world situations, additional images permit reconstruction of otherwise unobserved target facets permitting a more complete target reconstruction.

5.3 Case Study: SIFT Relationships and Scene Dependency

In the real world, perfect knowledge of the correspondences between cameras does not exist. Occlusions, specular reflections, and feature gradient similarities all frustrate the SfM/MVS process by limiting the knowledge of all features within each image. In fact the introduction of these effects so profoundly influences the final reconstruction, a separate case study is provided to reveal the inherent difficulties in extracting and matching features from real world scenes. The MATLAB simulation was unhindered by these real world effects so the Blender and Sadr City models were utilized to observe the effects of viewing geometry on the number of correspondences. This particular case study is only concerned with the effect of viewing geometry on correspondences and introduces the concept of scene dependency. The accuracy and completeness of the reconstructions from the various flight profiles will be addressed in later sections.

It is theorized correspondences between images will decrease as convergence angle increases using current feature matching techniques. Since image correspondences provide the basis for 3-D vertices in the final reconstruction, the quantity of correct correspondences directly impacts the model's likeness to the original scene. A special data collection file was used to investigate the effects of scene dependency, `linear_dynamic_-15D1000_300images`, which is uniquely suited to the effort since 300 images were obtained as opposed to the default of 50 images. This ensures high accuracy in selecting the images which closely match the desired convergence angle. For example, with 300 images, the error incurred by selecting two images which correspond to the desired convergence angle is only ±0.8° versus ±4.6° when only 50 images were collected.

As an illustration, Figure 42 depicts the features extracted from two images from different vantage points. The left image was acquired at the closest point to the target whereas the right image portrays the recorded scene from the furthest point. Each of these images show SIFT's ability to extract numerous keypoints from each image with 5323 and 10445 keypoints extracted from each image respectively.

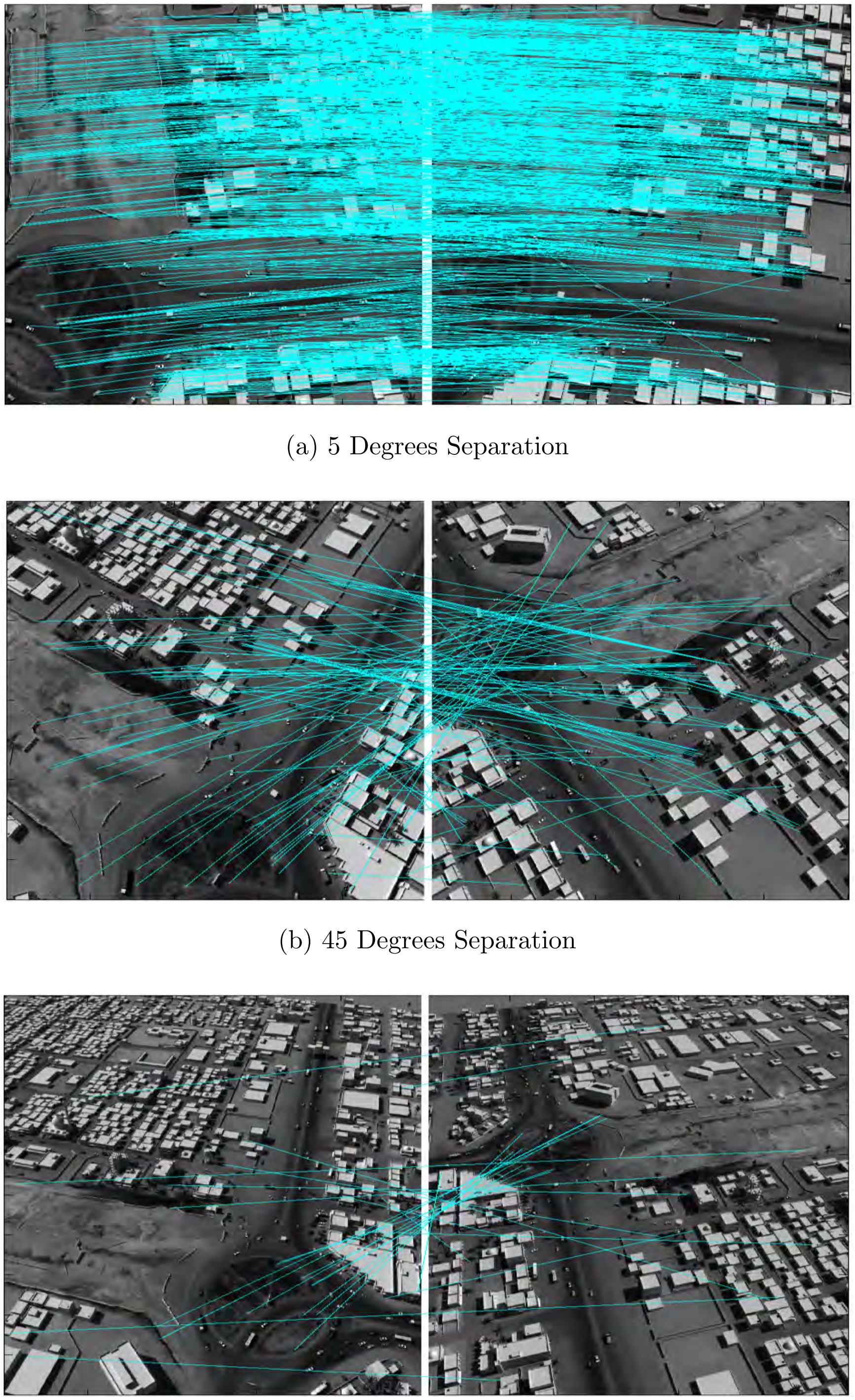


(a) Close proximity to target

(b) Distant to target

Figure 42. Keypoint descriptors identified in each image.

Obviously SIFT's extraction of numerous keypoints is irrelevant if the keypoints fail to match with another keypoint in subsequent images. Therefore, what remains to be seen is the effect of convergence angle on the number of correspondences. Figure 43 illustrates the drastic decrease in correspondences as convergence angle increases.



(a) 5 Degrees Separation

(b) 45 Degrees Separation

(c) 90 Degrees Separation

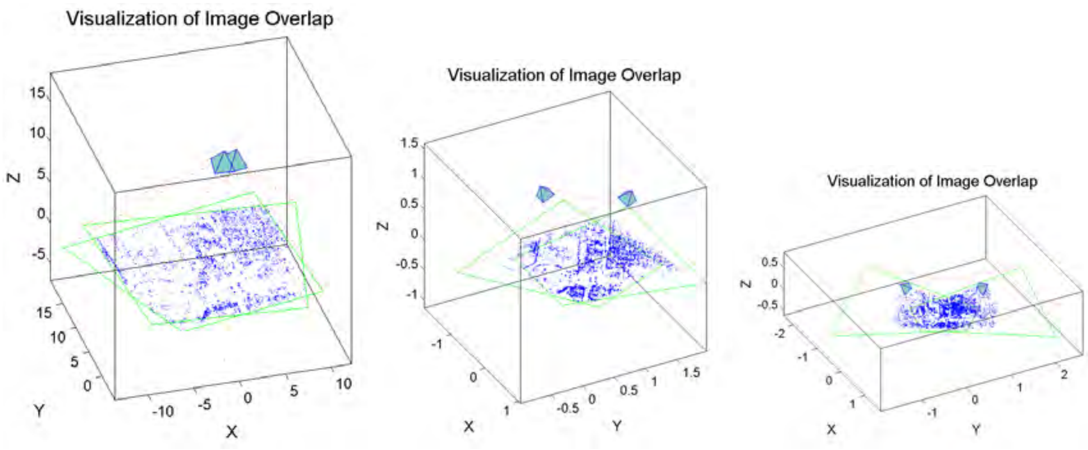
Figure 43. Correspondences between images at varying convergence angles. The decrease in correspondences in relation to the convergence angle is due to the reduced image overlap area and profound differences in image rotation and translation. Each of the subfigures above consists of two images separated by the convergence angle denoted. The cyan lines represent successful matches between the two images.

At a 5° convergence angle the baseline between the images is relatively short, and therefore the images acquired represent very similar projections of the target area. With nearly identical images, each keypoint has a high probability of being found in the opposing image due to the significant overlap and relatively small rotation and translation between camera frames. However as the convergence angle increases the number of correspondences decreases despite the increase in extracted keypoints in each image. Table 3 shows the relationship between the convergence angle and effect on correspondences.

Table 3. Number of keypoints extracted from each image at variable convergence angles. Note the direct relationship between convergence angle and correspondences and the inverse relationship to the number of matched keypoints. For complete dataset see Figure 46.

Convergence Angle	Image 1 Keypoints	Image 2 Keypoints	Matched Keypoints	Correspondence Percent
5°	5323	5378	1201	22.5%
45°	6663	5977	116	1.94%
90°	10445	9726	29	0.30%

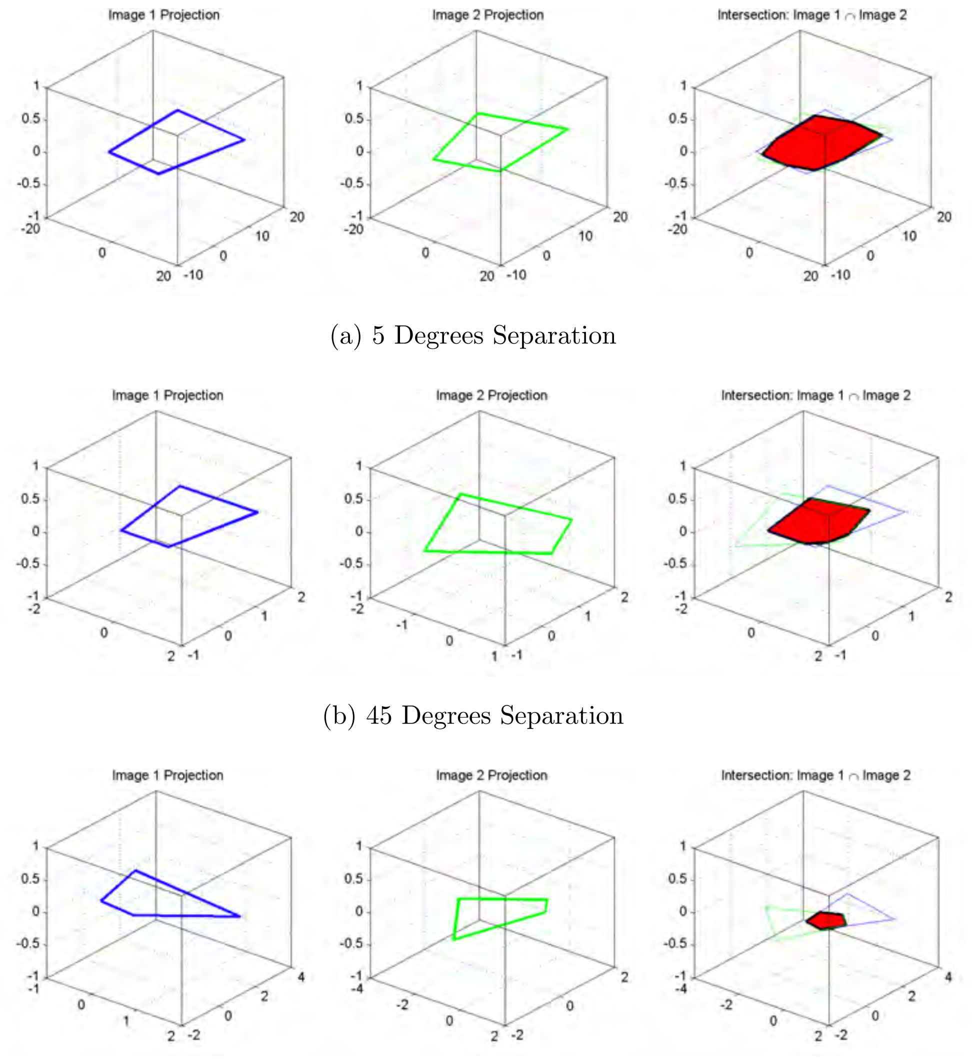
To explain this drastic decrease, two parameters are in effect; image overlap and rotation/translation between images. It is trivial to discuss the importance of image overlap to generate correspondences between images. Without commonalities between the images which occur in areas of overlap, it is impossible to expect a feature extraction and matching algorithm to register the two images. What is not so apparent is the relationship between image overlap at differing viewing angles. Figure 44 highlights this relationship by projecting the acquired image from each sensor onto the ground plane represented by the green polygons. The visualized sparse point cloud (blue vertices) and camera position and rotation were extracted from Bundler's `bundle.out` file.



(a) 5 Degrees Separation (b) 45 Degrees Separation (c) 90 Degrees Separation

Figure 44. Overlapping regions on ground plane. Note correspondences must be found within the overlapping regions. Also as the ground projection increases as seen in Figure 44(c), the relative ground sampling distance, or effective target resolution decreases. This leads to a proportional decrease in target correspondences.

Several significant features exist. First note how the preponderance of generated vertices occur within the overlapping region. The sparse point cloud was generated using all 300 images, and therefore vertices occurring outside the overlapping region are attributed to additional camera frames other than the two portrayed. Secondly the overlap between images decreases as the convergence angle increases, thereby reducing the possible number of correspondences. Figure 45 visualizes the reducing overlap between images. As look angle increases to the target, the ground projection becomes larger, yielding the increased number of keypoints seen in Table 3; however, the overlap significantly decreases resulting in rejection of many keypoints.



(a) 5 Degrees Separation

(b) 45 Degrees Separation

(c) 90 Degrees Separation

Figure 45. Relationship between the projections of the images and the overlapping regions highlighted in red.

When mapped over a wide range of convergence angles, the relationship between the number of correspondences and image overlap becomes apparent as seen in Figure 46. One immediately notes the exponential decrease in correspondences as the convergence angle increases whereas only a linear relationship exists between the overlap and convergence angle. This seemingly contradicts the previous relationship between SIFT correspondences and overlap, but scene effects especially occlusions have not been accounted for. Many of the keypoints derived from one image describe specific features within the scene, i.e. building edges, windows, and shadows. As the rotation and translation between the two images increases, the resulting viewing angle will also change. For example, in one image the east facing wall of a building is imaged whereas the west side is imaged at the far end of the collection. Obviously, the opposing sides of the building will not relate to one another. Therefore, correspondences between images separated by highly divergent viewing angles are reduced to common areas such as roof tops. Furthermore, the gradient feature descriptors change dramatically with large convergence angles frustrating gradient based feature matching algorithms. Even similar points such as those on the rooftops will be described by different gradients, and the points may not produce an image to image correspondence. In essence, the images will contain a large number of keypoints, but they will be of different features within the scene and contain a wide variety of feature gradients, thereby reducing image correspondences. This fact lies at the heart of scene dependency.

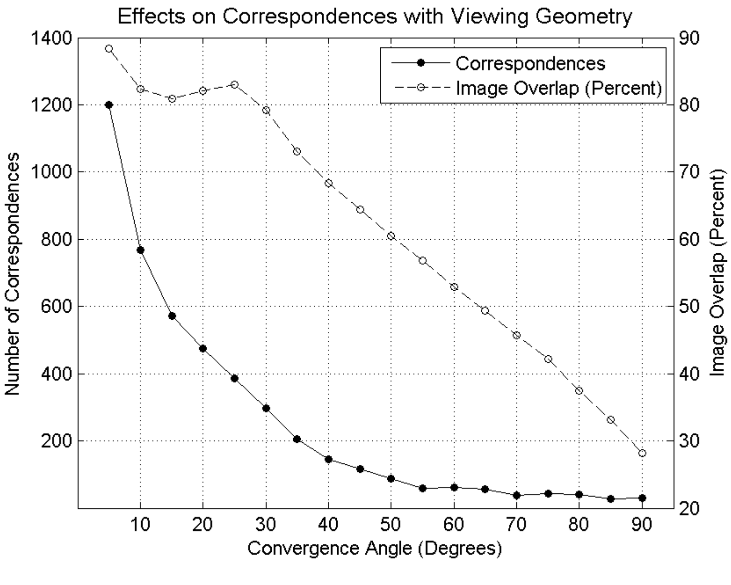
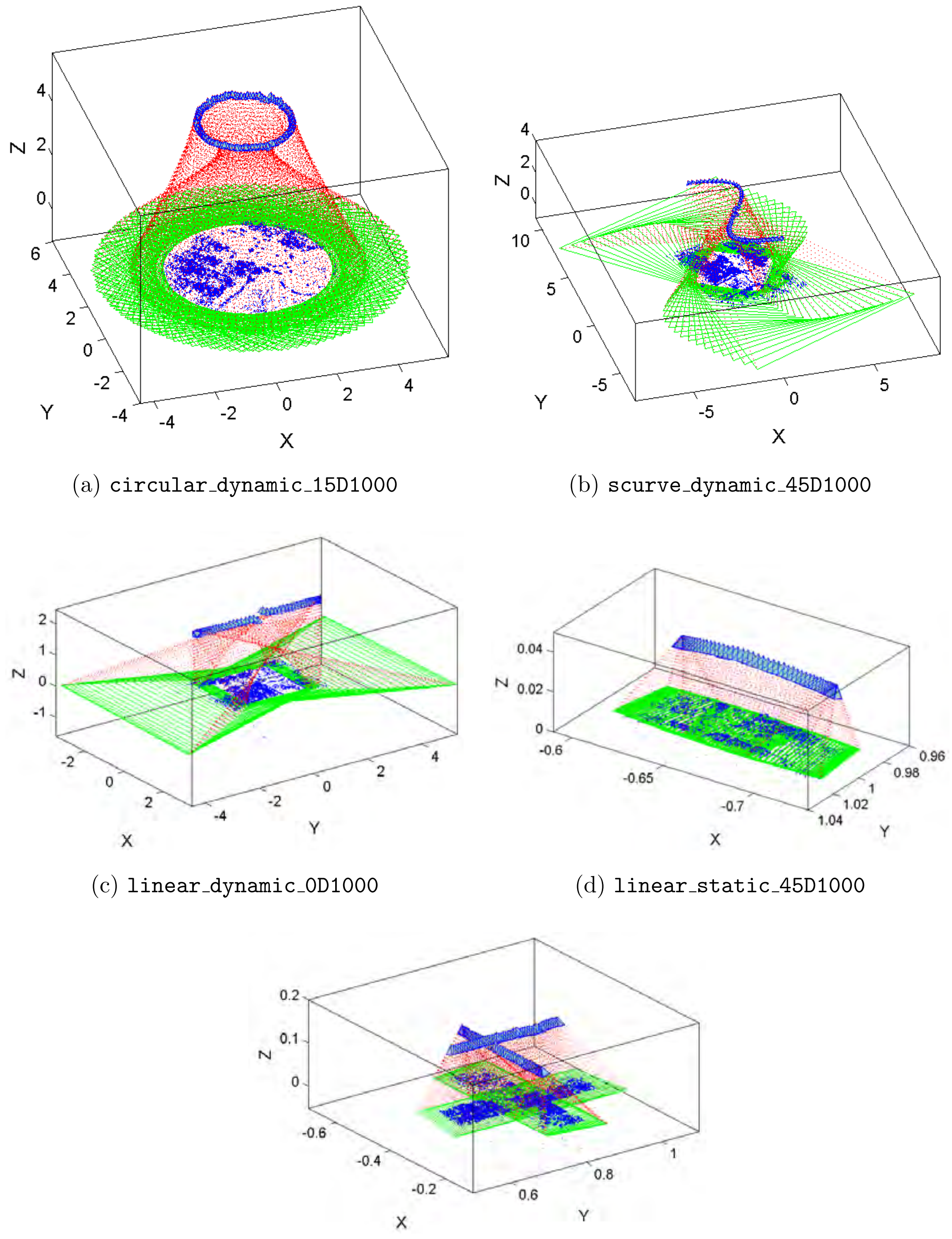


Figure 46. Number of correspondences as function of convergence angle.

It would be remiss to ignore the feature occurring at 25° in the above figure. It is expected image overlap will decrease as a function of convergence angle, but the data does not support this in its entirety. This feature is attributed to the fact the collection path did not occur directly at nadir over the target, but 15° to the north. Therefore at this angle the two ground projections are nearly orthogonal and the image 2 projection was nearly completely enclosed within the image 1 projection. A rapid misalignment of the two ground projections occurs slightly afterwards before resuming a linear decrease.

Finally as a precursor to the remaining analysis, Figure 47 and Figure 48 show the ground projections originating from the five main flight profiles represented by the green polygons. Both the camera pose and the sparse point cloud were extracted from the `bundle.out` file. Figure 47 depicts the ground projection with all camera fustrums and ground projections plotted whereas Figure 48 shows only selected camera frames for clarity. Note the preponderance of points occur within the image overlap area, a feature especially distinct in the circular orbit.



(a) `circular_dynamic_15D1000`

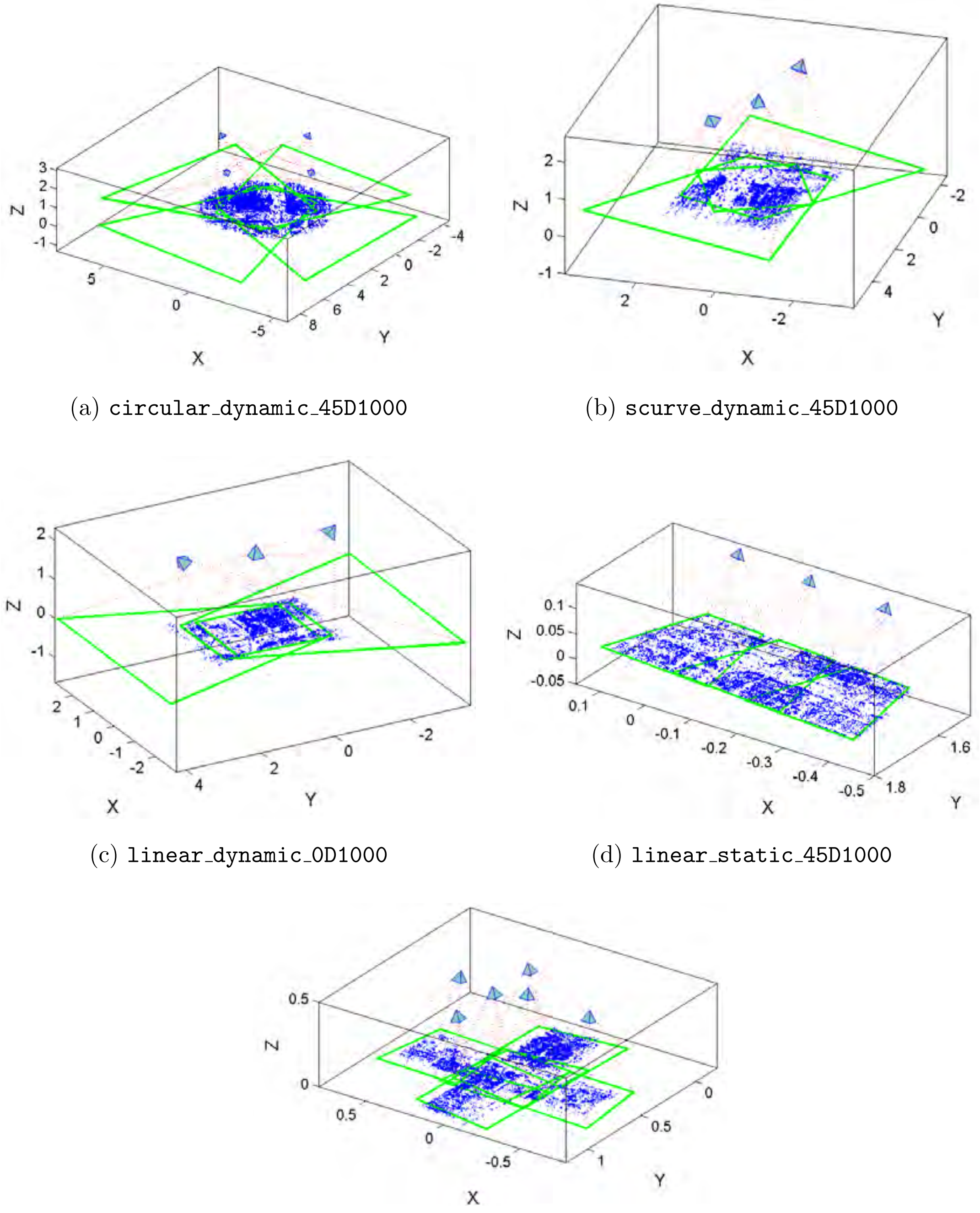
(b) `scurve_dynamic_45D1000`

(c) `linear_dynamic_0D1000`

(d) `linear_static_45D1000`

(e) `linear_staticcross_-15D-15H`

Figure 47. Ground projections from various reconnaissance profiles. Note in areas where significant overlapping occurs corresponds to the preponderance of image correspondences.



(a) `circular_dynamic_45D1000`

(b) `scurve_dynamic_45D1000`

(c) `linear_dynamic_0D1000`

(d) `linear_static_45D1000`

(e) `linear_staticcross_-15D-15H`

Figure 48. Only selected camera frames from the entire data set seen in Figure 47 are plotted to highlight the effect of camera orientation on ground projection.

The effects of convergence angle on correspondences was described in this section. It was observed the number of correspondences experiences an exponential decay with the increase of convergence angle. The decrease is attributed to both the reduction in overlap between images as well as occlusion effects resulting from the scene geometry. Cognizant of these principles it is now essential to investigate the effects of viewing geometry on reconstruction accuracy.

5.4 Flight Profiles: Reconstruction Accuracy (RMSDE)

The synthetic environment was successfully reconstructed as seen in the examples from the previous chapter. The task now turns to determining the accuracy of the reconstructions to the original model, hereafter referred to as the ground truth, seen in Figure 28. Complicating this effort is the relative nature which the SfM/MVS algorithms reconstruct the scene. Each reconstruction varies in terms of scaling as well as rotation about all three Cartesian axes. Although the vertices and camera parameters within the point cloud are mutually defined to one another, and therefore the model is self consistent, direct comparison of multiple point clouds requires correction of the scaling and rotation to a common parameter set. The same principle holds true for the ground truth model. Therefore, it is necessary to determine the appropriate scaling and rotation matrices to properly align the two point clouds for comparison.

The first task is to accurately determine the ground plane within the point cloud. Although several techniques are available, the author choose a RANSAC assisted method. As described in Section 2.5, a plane is fitted to three randomly chosen points within the cloud, and the distance of all other points to the plane is recorded. Those points whose distances fall within a predefined threshold distance are recorded as inliers. This process is repeated a statistically significant number of times and the plane with the largest number of inliers is kept. Upon conclusion the RANSAC process outputs the 3-D location of the three principle points (a, b, and c) as well as the coefficients of the plane equation $Ax + By + Cd = D$. The results of this process are seen in Figure 49 where the green triangle represents the plane with the largest number of inliers.

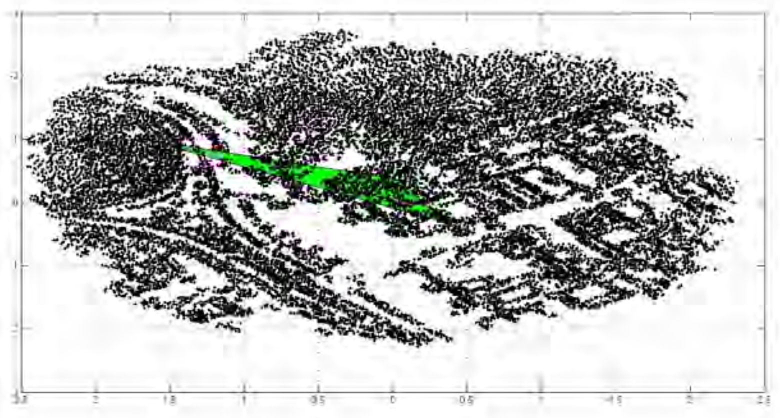


Figure 49. Determination of ground plane reconstruction. RANSAC fit parameters $t = 0.001$ where 277 of 18864 points fall within tolerance.

This process is suitable for this type of point cloud since the preponderance of points occur on a well defined ground plane. Caution must be exercised with applying this technique to a slanted or hilly terrain or when the reconstructed area contains proportionally equal number of points along difference axes. In such case an inaccurate ground plane may result.

The RANSAC process was applied to both the ground truth and reconstructed point clouds. The ground truth plane was determined by fitting 1,000,000 points to `sadrcity.obj` file to extract an extremely dense point cloud reconstruction of the area. Although the ground plane was initially assumed to be centered about the origin with a normal pointing directly along the z-axis, it was determined to have a slight inclination on the order of 10^{-7} in the dx and dy directions. This only slightly modified the next task to align the ground planes due to the relative reconstruction results. Using these three points (a, b, and c), two vectors, $\overrightarrow{ab}$ and $\overrightarrow{ac}$, were computed. The plane normal was then computed via cross product of the two vectors, $\overrightarrow{ab} \times \overrightarrow{ac}$, and verified against the RANSAC derived plane normal of (A, B, C).

The x and y rotations required to align the reconstructed normal to the ground truth normal was determined by calculating the tangent between the x and y coordinates. Once the deviation angles in the x and y directions were determined a standard 3-D rotation matrix was used to extract the necessary rotation matrix to correctly reorient to the reconstructed ground plane. The results are clearly seen in Figures 50 and 51 which demonstrate the successful rotation of the reconstruction to the xy plane.

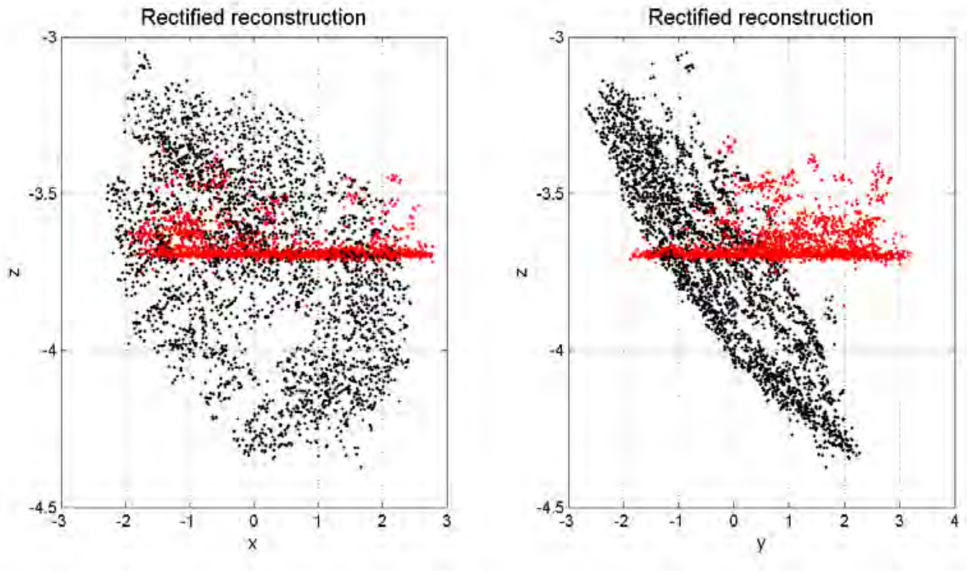


Figure 50. Corrected ground plane (red) compared to original reconstructed ground plane (black) in nominal situations.

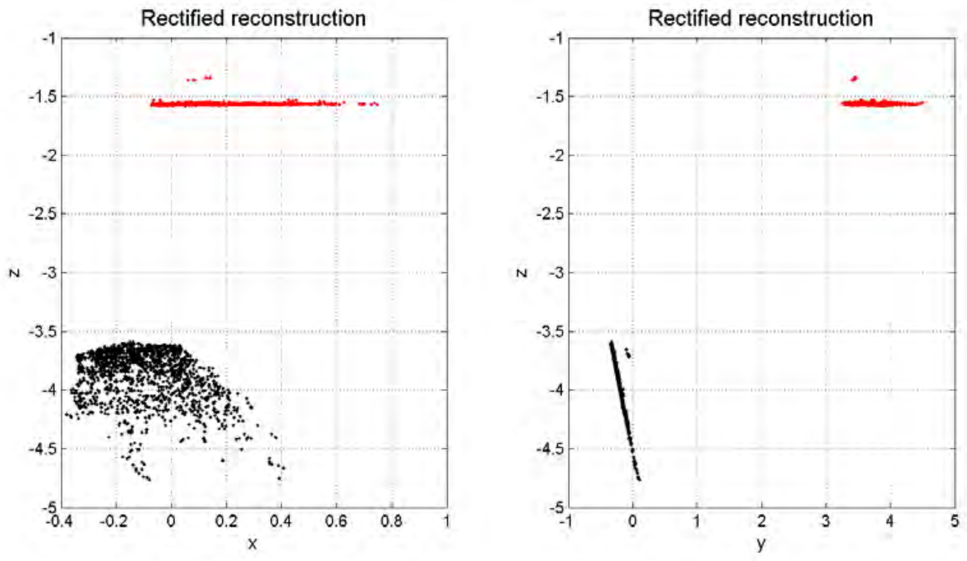


Figure 51. Corrected ground plane (red) compared to original reconstructed ground plane (black) in extreme situations.

Two observations are apparent in the above figures. First this technique applies equally well to both typical and atypical rotations as seen in Figure 50 and Figure 51. The second observation confirms an original assumption that most of the points exist along the ground plane. As clearly seen in the corrected reconstruction, the preponderance of points lie on a single plane.

The final product after a necessary translation of the entire point cloud to set the ground plane to $z = 0$ is seen in Figure 52. The results have been normalized to solve for the scaling difference between the ground truth and reconstructions. To this point the mutually self consistent reconstructions have been related to one another by solving for the scaling, x, and y rotational relationships seen in the figure below.

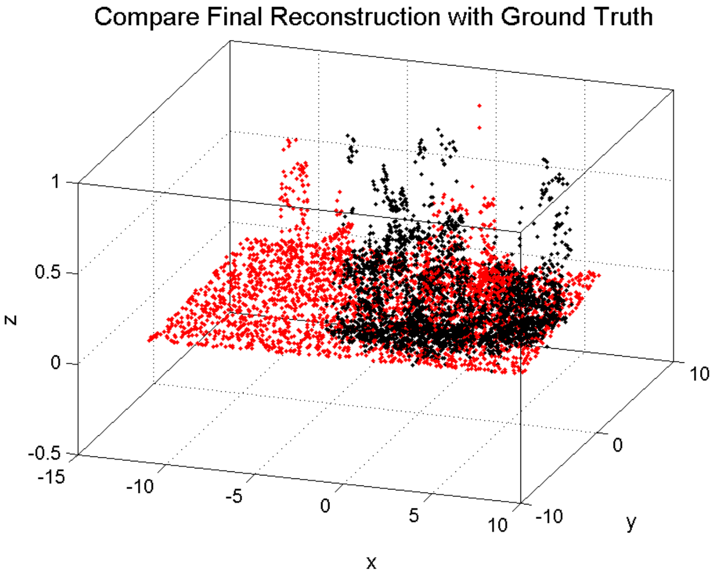
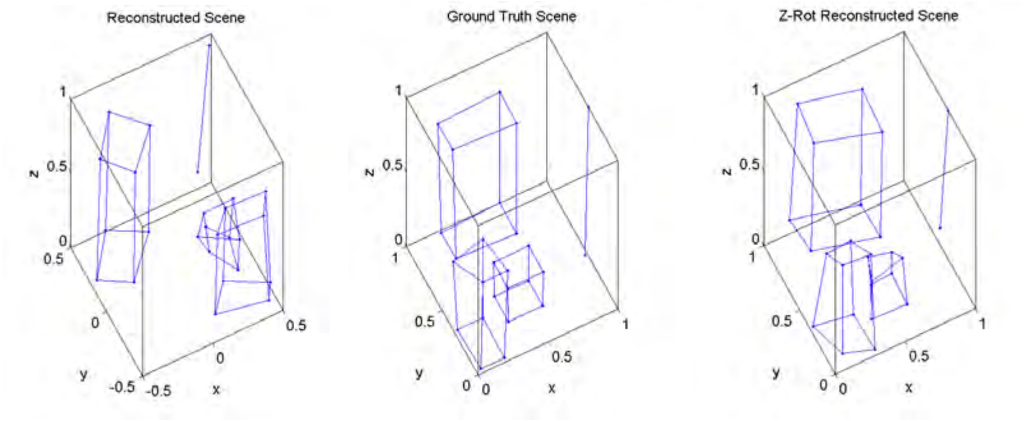


Figure 52. The ground truth point cloud representation (red) compared to rectified 3-D reconstruction (black). Note the x and y axis have been correctly solved but z-rotation remains.

The only remaining task was determining the necessary z-rotation to obtain the final alignment with the ground truth. This step required only the twenty six reference points and noting the right angle formed by the four points defining the the bottom of each structure to the x-axis. For instance, when a line intersecting the rear left and front left points was extended to the x-axis, it formed a 90° angle to that axis as seen in Figure 53(b). By averaging all such line combinations, the appropriate z-rotation was determined. The results of the final correction are seen in Figure 53(c).



(a) Reconstruction (b) Ground Truth (c) Corrected Z-Axis

Figure 53. Correction of z-axis rotation. The initial x and y axis correction seen in (a) still require z-rotation to align with ground truth (b). Vertices located along building bases form 90° to the x-axis. The final reconstruction (c) is corrected to all three axes rotations as well as normalized to scale with ground truth.

5.4.1 RMSDE: Linear Static Profiles.

The linear static profiles were analyzed for accuracy based on the alignment techniques discussed above. Once all flight profiles were correctly oriented to the truth model, the RMSDE was calculated using all twenty six reference points. The calculated RMSDE values, seen in Table 4 are provided in two forms, both absolute and normalized measures. The absolute values represent the actual RMSDE as calculated by Equation 49, whereas the second set of RMSDE have been normalized to the worst performer. The normalized values provide a global reference comparing the profile to all others. The lower the number the more favorably it performed in comparison to other reconstructions contained in this effort.

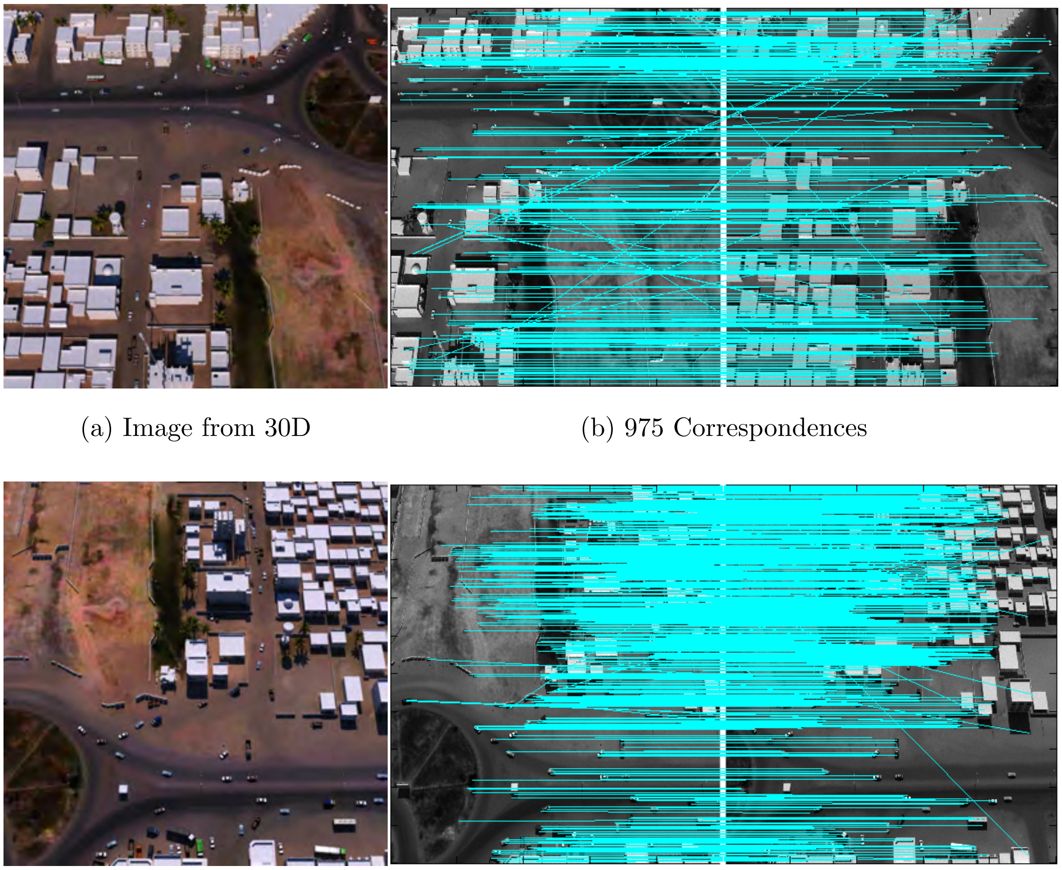
Table 4 clearly shows the intermittent difficulties the SfM/MVS algorithms have reconstructing the scene from the various look angles. At near nadir look angles, the algorithms failed to produce a satisfactory output. The term failed is subjectively applied to any reconstruction in which less than 1000 vertices were generated, and the target area is unrecognizable. For visual depictions of these failed reconstructions, see figures in Appendix B. The failed reconstructions at nadir and near nadir look angles correspond to an insufficient convergence angle and small number of camera frames. A static pass is limited to record images fixed to the airframe's velocity vector. Therefore the percentage of camera frames which contain the target area is dependent on the position and orientation of the camera as well as the camera's focal length. Under the conditions used in this effort, the number of camera frames containing the target ranged from 11 to 27 at look angles from 0° (nadir) to 60° respectively. Compare this to the 50 camera frames in which the target is in view for a linear dynamic flight profiles. The lack of camera frames is not the only issue. Due to the limited number of camera frames containing the target area, the convergence angle between the extreme

Table 4. Root Mean Square values for all linear static profiles.

Flight Profile	Absolute RMSDE	Normalized RMSDE
linear_static_-60D1000	0.0833	0.3695
linear_static_-45D1000	0.0580	0.2573
linear_static_-30D1000	0.0566	0.2513
linear_static_-15D1000	0.0672	0.2979
linear_static_0D1000	*Failed*	*Failed*
linear_static_15D1000	*Failed*	*Failed*
linear_static_30D1000	*Failed*	*Failed*
linear_static_45D1000	0.1654	0.7336
linear_static_60D1000	0.0981	0.4352
linear_static_-60H1000	0.0897	0.3979
linear_static_-45H1000	0.1300	0.5768
linear_static_-30H1000	0.0555	0.2462
linear_static_-15H1000	0.1050	0.4659
linear_static_0H1000	*Failed*	*Failed*
linear_static_15H1000	0.0897	0.3979
linear_static_30H1000	*Failed*	*Failed*
linear_static_45H1000	0.0663	0.2941
linear_static_60H1000	0.1747	0.7752

cameras is also restricted. At nadir the convergence angle containing all 11 camera frames was only 39.9° and at a look angle of 60° the convergence angle measured 40.0° with 21 camera frames containing the target area. Finally the richness of the data is fundamentally limited by the rotation degeneracy inherent in the linear static data collections. Since the camera is not rotating between sequential images, the rigid body motion lacks a necessary element and underconstrains the computation of the fundamental matrix.

Unlike the MATLAB simulation, real world factors complicate the accuracy of scene reconstruction beyond camera frames and convergence angles. This is seen as a disturbing trend in the linear static results. It is reasonable to expect similar results from symmetric flight profiles which share the identical number of camera frames and convergence angles, but as the RMSDE analysis indicates, `linear_static_15D1000` and `linear_static_30D1000` failed to produce realistic results whereas their symmetric counterparts performed well. This contradiction is attributed to scene dependency, or more specifically the correspondences between images. When comparing the correspondences between sequential image pairs from each symmetric profile, such as image 25 and image 26, the image pairs from the successful -30D pass contain significantly more correspondences than those from the failed 30D pass as seen in Figure 54.



(a) Image from 30D

(b) 975 Correspondences

(c) Image from -30D

(d) 1525 Correspondences

Figure 54. Scene effects substantially impact the resulting correspondences. The two images seen above were acquired similar distances and relation to the target; however at opposite viewing angles. The lack of correspondences from the `linear_static_30D1000` **significantly hindered the total reconstruction.**

This shows the inherent scene geometry dependency on the 3-D reconstruction beyond simple camera frames and convergence angles. Furthermore, the discrepancy grows as the convergence angle increases as seen in Table 5.

5.4.2 RMSDE: Linear Dynamic Profiles.

The linear dynamic reconstructions provided improved results compared to the linear static profiles discussed above. It is apparent the agile cameras, which allow for all 50 camera frames to image on the target area and the full 90° convergence an-

Table 5. Comparison of correspondences between image pairs from the symmetric linear_static_30D1000 and linear_static_-30D1000 data collections.

	Image Pairs			
Flight Profile	25\26	24\27	23\28	22\29
linear_static_30D1000	975	334	154	85
linear_static_-30D1000	1525	970	636	450

gle, drastically improve the resulting RMSDE. Furthermore since the camera images multiple sides of the target area, reconstruction of vertical surfaces is not limited to the parallax effects observed in the static collection. With increased visibility of the target, more facets of the structure may be reconstructed resulting in an overwhelming improvement in the RMSDE.

Table 6. Root Mean Square values for all linear dynamic profiles.

Flight Profile	Absolute RMSDE	Normalized RMSDE
linear_dynamic_-60D1000	0.1230	0.5456
linear_dynamic_-45D1000	0.0997	0.4421
linear_dynamic_-30D1000	0.1011	0.4487
linear_dynamic_-15D1000	0.0538	0.2387
linear_dynamic_0D1000	0.0310	0.1375
linear_dynamic_15D1000	0.0450	0.1997
linear_dynamic_30D1000	0.0809	0.3590
linear_dynamic_45D1000	0.1122	0.4976
linear_dynamic_60D1000	*Failed*	*Failed*
linear_dynamic_-60H1000	0.0811	0.3600
linear_dynamic_-45H1000	0.0880	0.3902
linear_dynamic_-30H1000	0.0545	0.2420
linear_dynamic_-15H1000	0.0602	0.2673
linear_dynamic_0H1000	0.0351	0.1556
linear_dynamic_15H1000	0.0449	0.1991
linear_dynamic_30H1000	0.0652	0.2894
linear_dynamic_45H1000	0.0685	0.3041
linear_dynamic_60H1000	0.0995	0.4412

The linear dynamic collections also reveal another important attribute to scene reconstruction; the contributions of target magnification, or spatial diversity, between images. While the camera center is far from the target area, the target reconstruction is small when compared to the collections acquired much closer to the objects. Not only has the usable camera frames increased and therefore convergence angles, but also the size of the target within each image. The combination of multiple perspectives has provided the rich camera motion necessary for accurate reconstructions.

A final trend worth mentioning is the gradual increase in RMSDE from nadir to the oblique look angles. As mentioned earlier, correspondences are required to generate 3-D vertices. At oblique look angles the imaged ground plane is large compared to the target area. Therefore target associated correspondences experience a substantial reduction as a function of distance to target. This feature is most prominently seen in the `linear_dynamic_60D1000` reconstructions seen in the appendix. Furthermore, occlusions from surrounding building structures or terrain increase as look angle increases. The combination of additional occlusions and the reduction in target prominence within the image projection lead to fewer target associated correspondences.

Graphically the linear static and dynamic results are seen in the following figure. Of note is the symmetric relationship observed in the dynamic collections which is absent in the static collection for reasons already discussed.

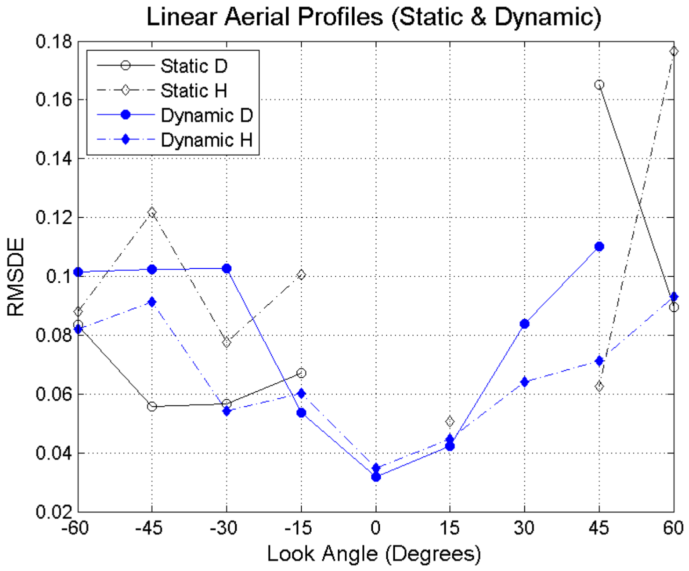


Figure 55. Effect of viewing geometry on both the static and dynamic linear flight profiles.

5.4.3 RMSDE: Multiple Linear Profiles.

Linear static profiles were limited to relatively few camera frames and convergence angles with the target in view. Therefore when multiple collections are combined improved results are expected due to the increase in image diversity. The combined passes yielded some of the most accurate results of this effort notably `linear_staticcross_-15D-15H` as seen in Table 7.

Table 7. Absolute Root Mean Square values for all linear static cross profiles.

-45H	-30H	-15H	0H	15H	30H	45H	
Failed		0.0675		0.1099		0.0973	45D
	0.0672		0.0356		*Failed*		30D
0.1323		0.1051		*Failed*		0.1165	15D
	0.0643		*Failed*		0.1530		0D
0.0503		0.0340		0.1426		*Failed*	-15D
	0.1872		0.0493		*Failed*		-30D
0.0761		0.0427		0.0494		0.0765	-45D

However, the combination of multiple linear passes also resulted in a significant number of failures. Most of these failures are attributed to individual pass failures for the aforementioned reasons, but this explanation cannot be universally applied. Instead profiles such as `linear_staticcross_45D45H` and `linear_staticcross_-15D45H` which consist of reasonably successful independent reconstructions, failed when combined. This is attributed to the substantial error associated in areas where the linear passes do not overlap. The SfM/MVS algorithms seek to determine the optimal arrangement of camera parameters and vertex locations during the bundle adjustment process and significant differences in the two individual passes complicate this effort. Restricting the SfM/MVS process to only those images containing the target area, yielded considerably better results as seen in Table 8. In fact `linear_staticcross_15D15H` which failed when the entire pass was evaluated, became the second best performer of the entire effort. Furthermore, the purely nadir passes which failed when analyzed independently and jointly reconstructed successfully when the data was limited to strictly the overlapping areas.

Table 8. Absolute Root Mean Square values for all linear static cross limited profiles.

Failed		*Failed*		0.0616		0.1678	45D
	0.1289		0.0833		*Failed*		30D
0.1833		0.0759		0.0322		0.0504	15D
	0.0364		0.0639		0.0552		0D
0.2254		0.0325		0.0424		0.0809	-15D
	0.0407		0.0458		0.0577		-30D
0.0623		0.0353		0.0421		0.0403	-45D
-45H	-30H	-15H	0H	15H	30H	45H	

The effects of orthogonal viewing angles is best seen in Figure 56. Reconstruction areas derived from overlapping profiles supplied superior accuracy compared to non-overlapping areas. Furthermore, when all images were included, competition between the four independent areas frustrated the reconstruction process often resulting in inferior results. When these non-overlapping areas were omitted, the RMSDE substantially improved.

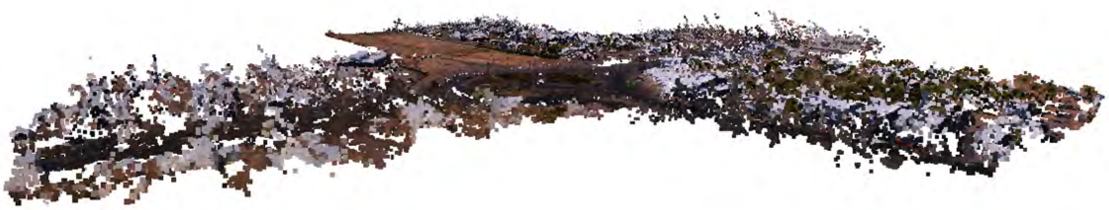


Figure 56. Substantially improvement in the RMSDE accuracy was observed when two orthogonal passes were combined. Furthermore note the substantial noise in the non-overlapping areas and reconstruction fidelity contained within the overlapping area.

At this point it is pertinent to introduce the impact of collection parameters on the reconstruction error ellipses. At narrow convergence angles the resulting depth error is larger compared to wider convergence angles. Figure 57 demonstrates this principle through four situations with a variable convergence angles, Φ, where the first image represents a static camera and the remaining a dynamically controlled camera. When orthogonal static passes are combined, the convergence angle is artificially increased by combining two camera frames from separate passes to represent the left schematic. This visual depiction of the error ellipses also portrays the trade off between convergence angle as defined by the baseline between images and the error. Dynamic passes are afforded the benefit of wide convergence angles to reduce depth error as well as narrow baselines to increase correspondences.

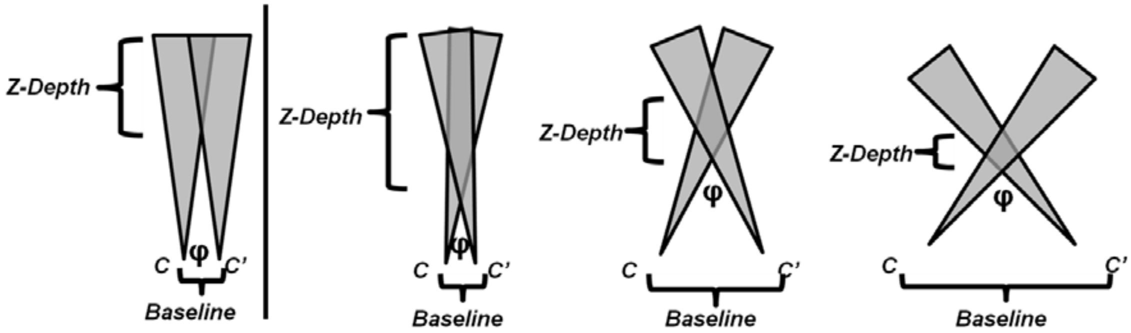


Figure 57. Error Ellipses with difference pass dynamics.

5.4.4 RMSDE: Circular Profiles.

Overall the circular profiles resulted in the most consistent RMSDE as seen in Table 9. The ability to image the target area from all sides allows for the reconstruction of vertices on all facets of the target buildings. Furthermore, the dynamic nature of the collection contributes numerous camera frames and wide convergence angles further increasing accuracy. These factors imply a circular orbit should provide the best reconstructions, but this is not true. The lack of translation with respect to the target limits target spatial diversity, and the reduced significance of the target within the global scene leading to eventual occlusion of vertices at oblique look angles adversely effects target reconstruction.

Table 9. Root Mean Square values for all circular profiles.

Flight Profile	Absolute RMSDE	Normalized RMSDE
`circular_dynamic_15D1000`	0.0431	0.1910
`circular_dynamic_30D1000`	0.0707	0.3135
`circular_dynamic_45D1000`	0.0647	0.2870
`circular_dynamic_60D1000`	0.1688	0.7489

The true limiting factor inherent to all circular profiles is motion degeneracy, or the lack of translation between camera frames similar to the rotation degeneracy seen in the linear static results. In other words, although the camera is physically translating around the structure when perceived from the scene's point of view, it could also appear the scene is simply rotating from the camera's point of view. Therefore an ambiguity exists in that the theoretical framework assumed the camera would translate relative to the object from frame to frame resulting in a real baseline and convergence angle. Despite this absence of translation, the plethora of correspondences between sequential camera frames allow for the computation of unique camera projection matrices, Π_1, Π_2, $\ldots$, Π_n. Regardless, the fundamental matrix will continue to be underconstrained by the data under pure rotational motions [34]. As such, Torr et. al. explained potential correspondences will be omitted and potential mismatches included, thereby limiting reconstruction ability.

In summary, circular profiles present an interesting dilemma. Due to the pure rotation and underconstrained fundamental matrix, a circular orbit requires more correspondences than other flight profiles explaining why this data set contained 100 images was frequently outperformed by other flight profiles.

5.4.5 RMSDE: S-Curve Profiles.

The s-curve profile contains all the necessary elements for a successful reconstruction: dynamic imaging, 360° visibility of the target, avoidance of motion and rotation degeneracy, and variable scaling of the target. The culmination of all these factors result in the richest data set analyzed as the diversity in correspondences provides ample data to compute an extremely robust and accurate fundamental matrix. The results of the RMSDE analysis are seen in the table below which reflects the accuracy to which the fundamental matrix was computed.

The reconstruction failure at 60° is attributed to the simulated keyhole encountered with azimuthal-elevation (AZ-EL) camera mounts where the mechanics of the

Table 10. Root Mean Square values for all s-curve profiles. Note scurve_dynamic_30D1000 provided the most accurate results.

Flight Profile	Absolute RMSDE	Normalized RMSDE
scurve_dynamic_15D1000	0.0502	0.2228
scurve_dynamic_30D1000	0.0232	0.1028
scurve_dynamic_45D1000	0.0429	0.1904
scurve_dynamic_60D1000	*Failed*	*Failed*

mount prohibit tracking of targets as they pass directly below. As the airframe passes directly overhead, the camera slews to the target until it reaches its maximum deflection of 90°. The camera mount must instantaneously rotate 180° degrees to continue tracking the object as the airframes moves away. The dynamic camera mounts simulated in Blender suffer from this limitation and the resulting images instantaneously flip 180° as the camera passes directly overhead to maintain a sky-up orientation. The effect on reconstruction is seen in the camera positions plotted in the sparse point cloud. In all s-curve flight profiles, the camera passes directly overhead, and the two camera frames located on either side of the pure nadir point are severely rotated from one another. This rotation hinders the reconstruction process which must account for the drastic change in rotation over a relatively short baseline. This manifests itself in the poor reconstruction of the camera pose at near nadir points. The degradation in rigid body reconstruction of the camera centers eventually reaches a peak at the 60° pass when the algorithm fails to extract forward camera movement beyond the nadir point. Finally the circular and s-curve RMSDE results are visualized in Figure 58.

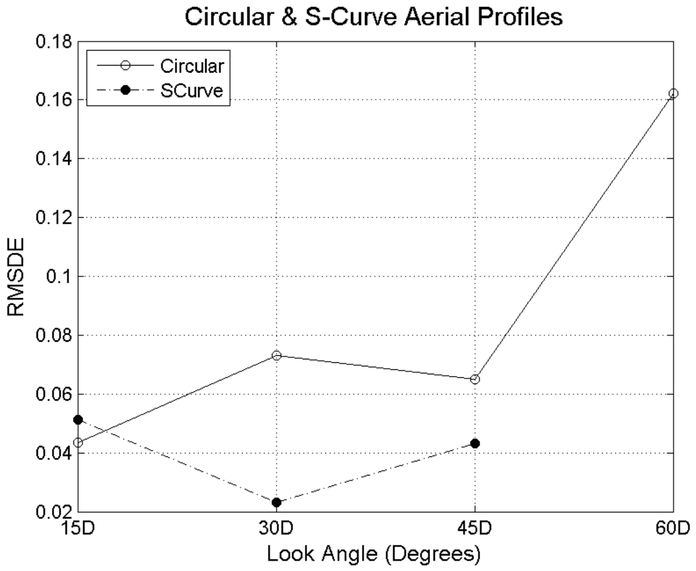
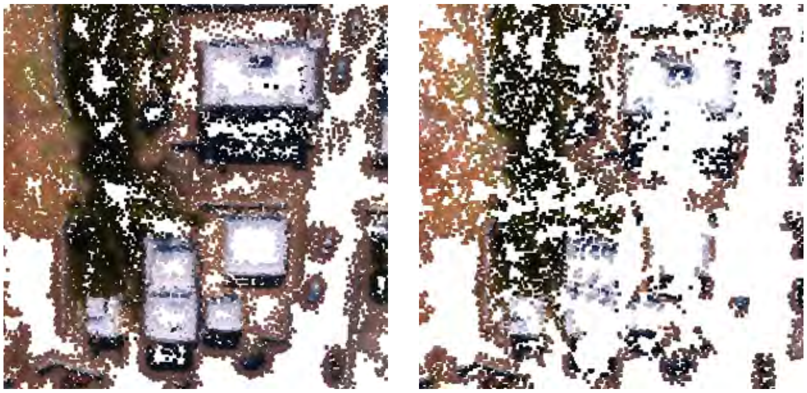


Figure 58. Effect of viewing geometry on both the circular and s-curve flight profiles.

5.5 Localized Point Density

The completeness of the reconstructions was analyzed to determine the degree to which the point cloud reconstructs all surfaces. This second measure of quality is required since a reconstruction with significant areas void of vertices may conceal important scene information. The two reconstructions in Figure 59 illustrate this point. The left image contains roughly five times the numbers of vertices but in terms of RMSDE, it is marginally poorer than the right. Therefore, the case is clear an additional factor is required to determine which aerial profiles render the most accurate and complete results.

A localized point density of each flight profile was determined by calculating the number of pixels within the immediate area surrounding the target area. A global point density measurement is ill suited since the variable look angles present different ground projections. For instance, many aerial profiles with look angles greater than 45° failed to accurately reconstruct the target area, but the total number of vertices



(a) High Point Coverage (b) Low Point Coverage

Figure 59. Localized density of target area of two reconstructions with similar accuracy but significantly different point densities. The left image contains 19783 points with an absolute RMSDE of 0.0707 whereas the right image contains 4807 points but an improved absolute RMSDE of 0.0672.

are two to three times greater than those reconstructions of the more accurate shallow look angles.

The localized point density for all linear profiles are displayed in Figure 60. Immediately apparent is the relationship between the density of points and look angle with the exception of the linear static profiles for reasons mentioned in Table 5 concerning scene dependency. Furthermore, the point density relationship to look angle seen in Figure 60 corresponds well to relationship seen when RMSDE is compared to look angle. During that analysis, it was determined the near nadir look angles provided highly accurate reconstructions for the linear dynamic profiles, whereas the linear static profiles struggled to produce identifiable results. The RMSDE relate to the density of points as seen in the figure. At near nadir look angles linear dynamic profiles exhibits the highest localized point density. Furthermore, linear profiles which failed to reconstruct obviously have the lowest point densities.

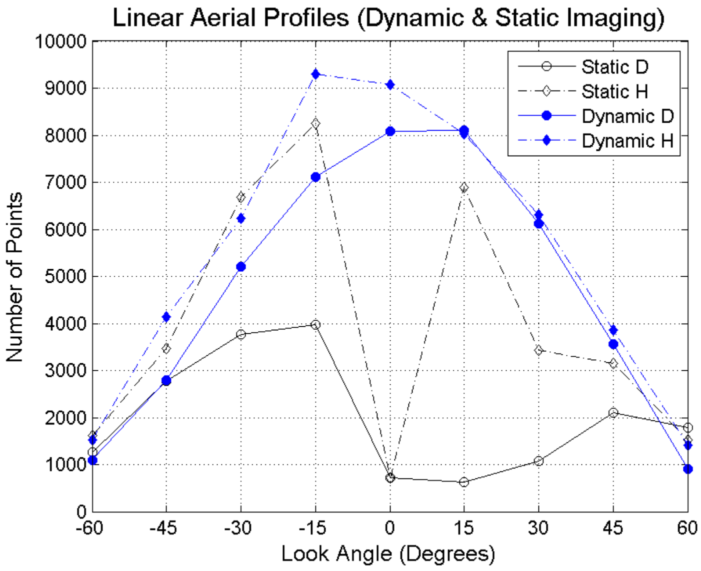


Figure 60. Localized density of linear profiles.

The circular and s-curve profiles also exhibit the same inverse relationships. At look angles demonstrating high accuracy, the density of points is proportionally elevated to other look angles. On the surface this statement seems trivial. However it reveals the interdependent nature of accuracy, density of points, determination of camera motion, and finally richness of image collection. Rich image collections allow for the extremely accurate reconstruction of the camera projection matrices. These projection matrices define the epipolar relationships between cameras by establishing the epipolar points and lines. The accuracy of the epipolar line calculation directly relates to the quantity and quality of additional correspondences found. The additional correspondences lead to more reconstructed 3-D vertices and therefore higher possibility of reconstruction accuracy.

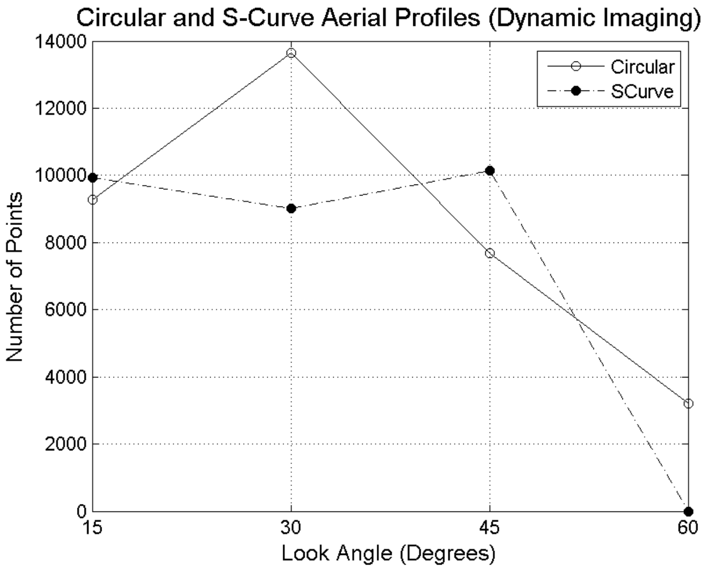
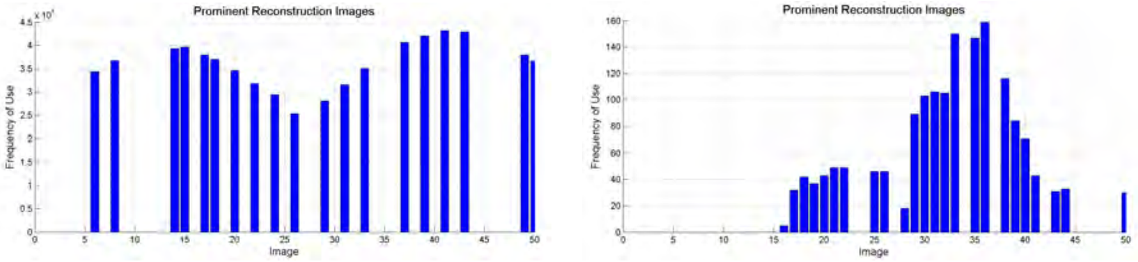


Figure 61. Localized density of circular and s-curve flight profiles.

5.6 Principle Images

Throughout this effort the criticality of camera frames has remained of pivotal importance. The camera frames supply the correspondences which lead to eventual 3-D vertices, and PMVS2 outputs include not only the reconstructed vertices but also the images in which the patch was calculated from. A brief investigation into which images were used in the reconstruction was accomplished to determine if the number of images or convergence angles could be reduced without negatively effecting the reconstruction accuracy and the interplay involved when clustering images for parallel processing.

Upon preliminary analysis, a distinct and unexpected trend was observed. Flight profiles exhibiting positive attributes, such as highly accurate results and vertex rich point clouds, shared a common multi-modal structure dependent on image distribution and frequency of use. Figure 62 is a histogram of the 50 input images identifying both the frequency and individual camera frames used in the reconstruction. Aside from the multi-modal structure, the distribution reveals which images were omitted during the dense reconstruction process. These images are removed from consideration by CMVS since they are nearly identical and redundant images contribute no additional information to the solution, increase computation time, and the short baseline between images leads to significant z-depth errors. It is important to note the sparse reconstruction process employs the entire data set and solves for all camera matrices whereas the image distributions seen below are representative of a dense point cloud.



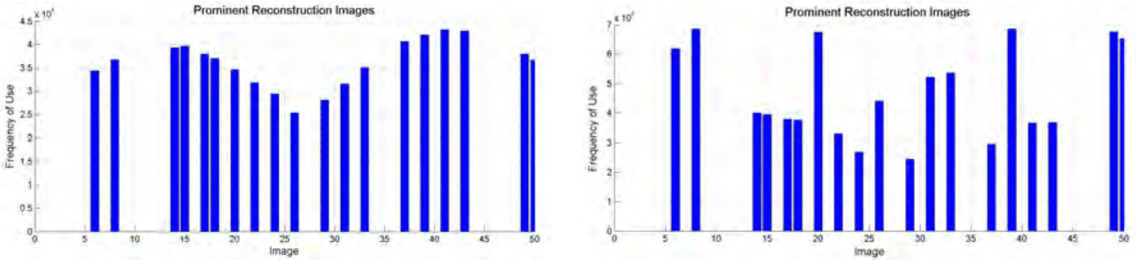
(a) Highly Accurate Reconstruction

(b) Poor Reconstruction

Figure 62. Principle images from excellent reconstructions exhibit multi-modal image distributions (a) which is absent in poorly reconstructed images (b). Note low image use in poorly constructed images compared to the high image use in the well behaved counterparts.

All profiles were analyzed, and the multi-modal structure was confirmed to be a property of all successful reconstructions. Furthermore, the circular profiles exhibited similar multi-modal structure even though all camera centers were equally spaced, target oriented, and identical distances from the target. The multi-modal structure can be explained by observing specific characteristics of each pass. First in both the linear and s-curve profiles, the pass can be segregated into pre and post-nadir camera orientation groups. At this point an important distinction is required. Images groups refer to collections of images which share similar pairwise photometric scores whereas the term clusters is reserved for collection of images within the CMVS process. The images contained within each group portray a different orientation to the target area and therefore the images will relate more closely to one another than images in a different group. Supporting this argument is the location of the minima between each of the modes. In all linear and s-curve scenarios, the local minima occurs at roughly image 25 or the image nearest to nadir segregating the pre and post-nadir image groups. During reconstruction, PMVS2 segregates images into groups denoted by $V(p)$ in which the patch, or vertex p, is visible as determined by Equation 39. Since the two major patch groups, pre and post-nadir, would clearly contain different different $V(p)$'s, or image sets, two modes would be expected. The circular passes would experience similar grouping of images associated with patches at different locations in the target areas. Furthermore, the Gaussian structure within each mode can be explained by a simple fact; images acquired in the middle of the pre and post-nadir pass segments will more closely relate to images of that segment. Knowledge of where the principle images occur and their relationship to the overall reconstruction would aid immeasurably to ensure the right images are collected.

In the above explanation, it was assumed PMVS2 has visibility to all images within the data set; however, this may not be true. CMVS properly eliminates redundant images and clusters the remaining images into similar image sets. It was already demonstrated in the above example, 24 of the 50 images were retained while the remainder were rejected based on redundancy. A second data conditioning step divides the remaining images into clusters as to not violate the maximum cluster size. As stated in Table 2, the default value for maximum image cluster size was 30 images. In Figure 62, the default value was used but after elimination of similar images only 20 remained, well within the size constraint. To verify this hypothesis a CMVS/PMVS2 was rerun on the data with the maximum cluster size changed from 30 to 10. Under these conditions, 4 to 5 clusters would be expected. As seen in Figure 63, overconstrained CMVS parameters can negate the ability of PMVS to group images based on patch visibility.



(a) 30 images per cluster

(b) 10 images per cluster

Figure 63. Relationship between the Maximum Cluster Size and multi-modal structure. The reduction of maximum cluster size allows for more clusters which manifest themselves as additional modes on the image distribution charts.

When a maximum cluster size is selected which requires CMVS to produce two clusters, additional insight is revealed. In Figure 64, a maximum cluster size of 14 images was enforced. This required CMVS to cluster the 26 remaining images into two distinct clusters. This figure supports both arguments above in that the multi-modal structures is dependent on both the maximum cluster size and PMVS2's allocation of images into $V(p)$ defined image sets. Again the dual mode structure is seen; however, now that two clusters are used as opposed to the original one cluster, the distribution of images between clusters is seen in the red and blue bars. CMVS accurately split the image set into clusters with similar viewing perspectives while PMVS2 weights the central images to each cluster more than the others. The final observation available is the overlap between the clusters occurring at their intersection. A small degree of cluster overlap is required to ensure each cluster may be registered with another upon the conclusion of parallel processing of each cluster. This overlap is directly analogous to the image overlap required to generate image correspondences.

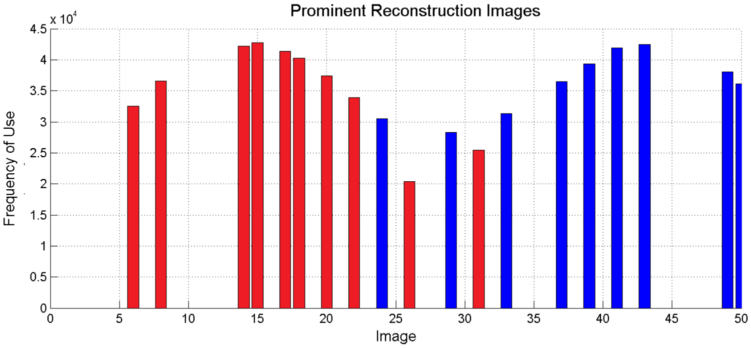


Figure 64. Distribution of images between two clusters of max 14 images. The red and blue bars represents each of the two clusters and the overlap occurring between the two is required for final rectification of the two independently processed clusters.

Furthermore, the angle between the individual bimodal peaks occurring in the linear flight profiles with a maximum cluster size of 30 was determined and plotted against look angle as seen in Figure 65.

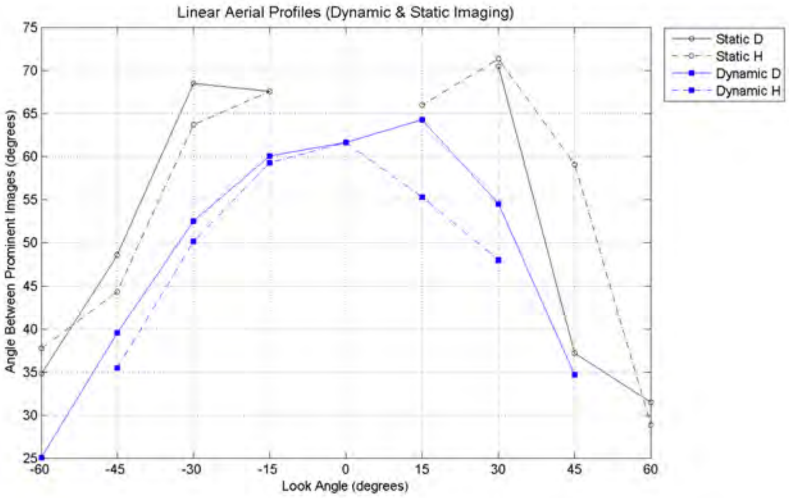


Figure 65. Principle Images used in linear passes. Note as angle from nadir increases separation between primary images decreases.

The graph reveals a trend common to all linear reconstructions: as look angle deviates from nadir the angular separation between principle images decreases. This can be explained by the gradual elimination of the pre and post-nadir point. At pure nadir there is a clear distinction in image sets as each images independent sides of the target area. As the look angle increases the prominence of this distinction fades as more and more images are viewing a common side of the structure. Therefore as look angle increases the distinction between images is reduced and all images are more likely to be grouped together with the center image sharing the most commonalities to all other images.

5.7 Case Study: Transformation to Real World Coordinate Systems

Inclusion of the camera's IOPs and EOPs results into the SfM/MVS reconstruction process results in a metric reconstruction with real world coordinates which provides an avenue for a direct comparison to existing LIDAR 3-D data. This case study presents preliminarily efforts in which this data was used to perform a coordinate transformation from the relative SfM coordinates to real world. This endeavor would be best performed during the SfM process, unfortunately Bundler and CMVS/PMVS2 do not allow insertion of known parameters with exception of focal length as measured in pixels.

The relationship, $x' = Hx$, previously related images, but it may also be used to relate disparate datasets such as SfM/MVS and LIDAR 3-D reconstructions. Again, the CLIF II 2007 and LIDAR data used in the first case study supply the necessary information. The first task entails determining the correspondences between the datasets required for accurate registration. Although no direct correspondences between the LIDAR and SfM datasets exist, the actual camera's EOPs were available which act as substitutes for the real world points contained within the LIDAR dataset. In such case the camera's EOPs (x) allow transformation of the relative SfM derived camera EOPs (x') to real world coordinates. Walli showed the pseudo-inverse solution, seen below, provides the necessary H to register the disparate datasets [35].

$$H = x'x^T \left(xx^T\right)^{-1} \tag{51}$$

The solution should relate the entire SfM reconstruction to real world coordinates allowing the direct comparison to LIDAR data. Transformation results from SfM camera coordinates to real world is seen in Figure 66 where it is immediately apparent the transformation was successful. The original SfM coordinates, including camera centers and scene vertices, were normalized to values between 0 and 1, and after the transformation the camera centers were accurately projected to the real world locations. Also note the fidelity of the SfM derived camera centers. The SfM process can accurately identify the relative motion of the camera, however, not to the fidelity of the true points.

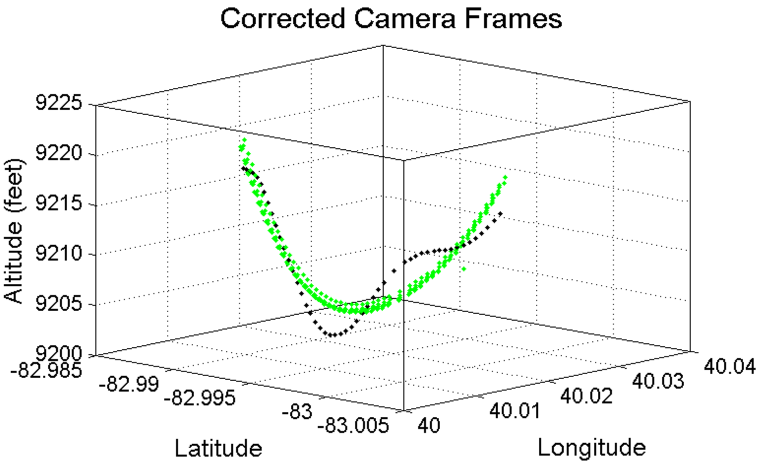


Figure 66. Transformation of SfM camera positions (green) to real world coordinates (black). Note minute variation in elevation as the collection airframe moved between successive images.

Thus far the analysis has centered about the camera centers since they provided the required correspondences between the datasets. Unfortunately the homography matrix did not accurately project the SfM scene vertices to accurate real world coordinates as seen in Figure 67.

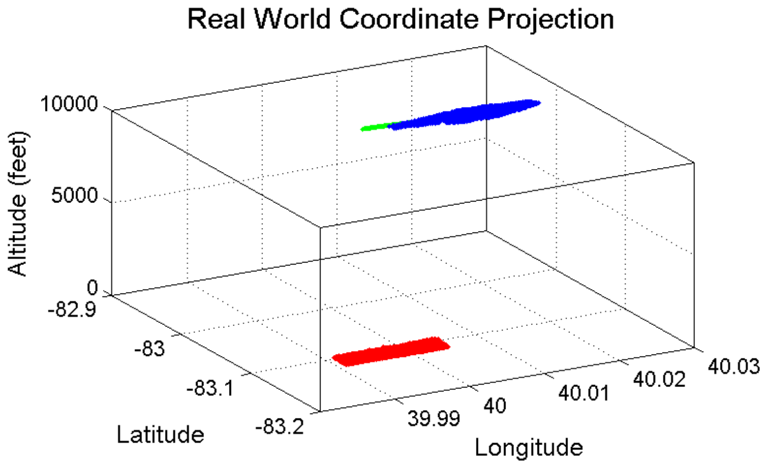


Figure 67. Transformation of all SfM vertices (blue) to real world coordinates in comparison to the LIDAR derived point cloud (red). The use of localized correspondences results in a poor transformation of the global scene.

The failure of the homography matrix to accurately transform all SfM points is twofold. First only correspondences between the SfM and real world coordinates were sourced from an extremely small dataset unique to all SfM vertices. Limiting the correspondences in this manner only provides the data for that limited set. Secondly, derivation of the homography matrix requires correspondences amongst all three dimensions to accurately register two 3-D datasets. In this situation correspondences only exist on a planar surface. The combination of these two factors only allows for the correct transformation of the correspondences themselves and not for the global scene. Since additional correspondences are unavailable, a method must be determined to allow direct insertion of the camera's EOPs directly into the SfM process. To date this method remains elusive, but remains a path for active future research.

5.8 Operational Considerations and Impact

This effort's salient feature is the applicability of laboratory models to real world operations. The ability to create realistic 3-D scenes without complex LIDAR systems is of immense importance to the warfighter. Not only will this technology open new opportunities but also augment or replace more expensive solutions. In today's budget constraints this attribute may be the most important. Therefore a brief discussion surmising the required flight parameters for a high quality 3-D reconstruction is required.

5.8.1 linear_static.

Linear flight profiles with fixed cameras are not a preferable method of data collection at near nadir collection geometries. The target area was frequently unidentifiable at near nadir passes and only marginally accurate at look angles greater than 45°. However this collection method is widely implemented due to its ease of installation and operation. In such case, the combination of orthogonal passes significantly improves the reconstruction, and by limiting the reconstruction to only areas of overlap, additional improvements may be realized. In fact, such data conditioning allows for successful reconstruction of purely nadir passes. In SfM/MVS algorithm terminology, the a linear profile with a static camera provides a meager data set with little variability in both spatial and angular diversity as well as structure visibility due to poor image overlap. Furthermore, these limitations reduce the maximum number of camera frames and convergence angle necessary for adequate reconstructions. Finally, the lack of rotation between the camera frames and scene introduces motion degeneracy and underconstrains the solution presenting a upper limit to the reconstruction accuracy.

5.8.2 linear_dynamic.

Linear dynamic profiles present well formed reconstructions with minimal effort. Reconstruction accuracy decreases as a function of increasing range and look angle so their use should be limited to nadir and near nadir data collections. Beyond look angles of 45°, the performance drops sufficiently to warrant transition to linear static flight geometries. Operationally, the linear passes require minimal resources to execute, and the addition of an agile sensor is not uncommon on modern surveillance assets. Computationally, the addition of a dynamic camera allows for sufficient diversity to eliminate all degeneracies and provide improvements to linear static collections. Furthermore, the agile sensor permits all images to contain the target increasing both the number of cameras and convergence angles.

5.8.3 circular_dynamic.

Circular dynamic flight profiles offer accurate and dense reconstructions similiar to linear dynamic profiles. The long dwell time necessary for the collection reduces the operational feasibility for such a collection. Furthermore, the apparent lack of translation introduces motion degeneracy presenting an upper limit to the reconstruction. However the 360° view of the target area allows for complete reconstruction of all building facets which deserves additional consideration in mission planning.

5.8.4 scurve_dynamic.

The s-curve profile offers the most diverse data set, free of degeneracy, and near 360° target visibility. For these reasons it offers the most accurate and dense reconstructions. Operationally, the pass benefits from a limited dwell time but high complexity. If these complexities can be surmounted a s-curve dynamic pass offers the highest accuracy reconstructions.

5.9 Chapter Summary

This chapter quantified the likeness of the 3-D reconstruction to the actual scene geometry for both controlled simulation results and synthetic real world environments. Within the MATLAB simulation, it was determined an increase in the number of cameras refines the estimated focal length which provides improved results. Furthermore, a direct relationship between improved camera pose and reconstruction accuracy was observed where both increased with convergence angle. Despite these relationships, an expansive minima was seen in the reconstruction accuracy which highlights the algorithm's robust ability to produce accurate 3-D reconstructions over a wide range of input parameters. Furthermore, several aerial profiles designed to mimic typical reconnaissance profiles were analyzed for RMSDE and completeness. Linear static profiles afforded the least accurate results at near nadir orientations; however, at oblique angles greater than 45°, this profile provided the most accurate reconstructions. On the other hand, linear dynamic profiles at near nadir provided extremely accurate results but experienced a substantial reduction in accuracy at oblique angles. Circular and s-curve profiles experienced comparable accuracy albeit at more complicated aerial profiles and increased computational requirements. The distribution of images within the algorithm provided insight into which images the SfM/MVS algorithms heavily rely. A multi-modal distribution is seen which transitions to a purely Gaussian distribution as look angle increases. Finally the transformation of SfM derived vertices from SfM relative coordinates to real world coordinates was pursued. The transformation unsuccessfully projected the SfM vertices back to real world coordinates due to the limited availability of correspondences between the disparate datasets. However if additional correspondences or an insertion point into the SfM process is determined the transformation to real world coordinates will be successful.

VI. Recommendations and Future Work

The quantification of reconstruction accuracy and completeness was examined in this effort. During the course of this effort additional research avenues were uncovered, but restrictions in time and resources did not permit their investigation.

The first area of additional research is continuation of the work previously accomplished on integrating actual camera IOP/EOP matrices into the reconstruction solution. Modern surveillance camera and aircraft sensor data can be fused to provide the exact placement of the camera in terms of latitude, longitude, altitude, yaw, pitch, and roll. This information coupled with foreknowledge of the sensor's focal length, provides all the necessary information to compute an exact camera projection matrix, Π_1, Π_2,..., Π_n. As seen in the MATLAB simulation slight deviations in the focal length and camera pose introduce significant errors into the 3-D point triangulation.

Secondly, the models used to determine the RMSDE involved the manual selection of points within the point cloud to represent each of the twenty six reference vertices. Unfortunately some aerial profiles did not image a specific vertex, therefore, a reconstruction is impossible. As a result, the closest vertex to inferred reference point was selected. A fundamental understanding of manmade objects is they appear as simple geometric shapes. Therefore, it should be feasible to fit shapes within the point cloud to represent the various structures. This allows for the exact determination of a building face if only the roof and ground plane are known. A much improved representation of the actual scene would result. Furthermore, this would provide the necessary step to implement an image derived reconstruction in higher order physics based simulator such as DIRSIG.

Thirdly, it was shown additional cameras and wide convergence angle improve results but at the cost of computing cycles and processing time. Therefore, a real-time reconstruction would require a minimal data set with sufficient information to retain the required degree of accuracy. Pursing the minimal information requirement for environmental sensing will pave the way for robotic vision. To support this effort, the minimum convergence angle, η found in Equation 1, must be determined to adequately relate the required baseline distance, approximate height of target, and target distance.

Finally, the natural evolution of this effort is application to real world data. Testing these principles against video footage from aerial vehicles would provide the final validation to justify the use of complicated flight paths and dynamic sensors.

VII. Major Accomplishments

This section identifies the distinct contributions of the author as a direct result of the research discussed in this thesis.

7.1 Effects of Variable Viewing Geometry on Reconstruction Accuracy

Within the MATLAB simulation, the influence of variable convergence angles and number of camera frames required on the accuracy of epipolar reconstructions was determined. Thresholds of 3 camera frames at a 6° convergence angle produce highly accurate results when perfect knowledge of the image correspondences is known. Subsequent addition of camera frames and an increase in convergence angle only marginally improve reconstruction accuracy. This dependency allows for tradeoffs between operational collection requirements and computer processing and the required accuracy. Furthermore, analysis of these dependencies showed a clear correlation between camera frames and convergence angle to estimated focal length and camera pose estimation respectively. This provides the nascent research to seed further research and direct additional investigations.

7.2 Effects of Operational Flight Profiles on 3-D Scene Reconstruction

Previous research entailed choreographed collections of images or massive Internet wide searches resulting in thousands of images. This effort explored the tradeoffs when image collection is limited to typical overhead reconnaissance flight profiles utilizing dynamic and static sensors. As a result the necessary connection from laboratory to field operations was established providing a basis by which operational test and evaluation planners can plan further testing and transition to operational use.

7.3 Existence of Prominent Images

The existence of bridge images within a data set has been documented; however, explanations into their origin stem from unification of disparate image sets. This research shows the bridge images do exist as seen in the overlap of imaging clusters and groups. However these bridge images are not the most prominently used images within the data set. That distinction is reserved for images which share similar pairwise photometric scores with other images in the collections. Reconstruction algorithms heavily rely upon these images for vertex generation and therefore their collection is paramount at the distinct look angles.

7.4 MATLAB Tools

A suite of MATLAB tools was generated in this effort encompassing a gamut of investigatory tools from software to generate and reconstruct simple 3-D objects to analytical tools to probe the cryptic output of the reconstruction algorithms. These tools include:

- Epipolar reconstruction from user created 3-D objects.
    - Allows for the generation of computer generated images with complete control of the number and location of cameras.
    - Produces projective, affine, and euclidean reconstruction models with RMSDE accuracy to base structure.
- Bundler and PMVS/CMVS data extraction tools.
- Extraction of sparse and dense point clouds as well as corresponding camera fustrums to MATLAB environment.
- Visualization and calculation of ground plane projections from cameras at SfM generated locations and orientations. Furthermore calculated image overlap between selected cameras or all cameras.
- Extraction of images used in the reconstruction of each vertex.

VIII. Conclusion

This effort explored the impacts of viewing geometry on 3-D scene reconstruction from 2-D imagery using Structure from Motion (SfM) and Multi-view Stereo (MVS) reconstruction techniques. As with all data processing and reconstruction algorithms the quality and fidelity of input data has a direct effect on the results and the epipolar techniques used were no exception. The salient goal of this effort was to determine optimal flight profiles necessary for accurate 3-D reconstruction of real world targets. However, such an effort requires foreknowledge of the SfM and MVS reconstruction algorithms. For this reason, a multi-phase effort into the viewing effects on 3-D reconstruction was pursued. The first phase of the effort required the implementation of a fully programmable software package in which image generation and scene reconstruction can be strictly controlled. Leveraging the knowledge gained from this effort, the second phase used academic 3-D reconstruction software packages to reconstruct a typical urban scene based on images acquired from typical airborne flight reconnaissance profiles.

In support of the first phase, a high fidelity computational simulation was successfully implemented within the MATLAB environment. The simulation introduced a simple 3-D structure in which a series of images at variable spatial positions and rotational orientations were acquired. By varying the number of camera frames from 2 to 20 and convergence angles from 1° to 100° in 1° increments, the relationship between the number of camera frames and convergence angle on reconstruction accuracy was determined. The reconstruction technique allows observation of a lower threshold required in which 3 or more cameras were required at a convergence angle of 6°. With only 2 cameras the extracted focal length frequently assumed imaginary or null values despite an initial guess of 400 mm. As the number of camera frames increased, the algorithm ability to determine the actual focal length improved eventually reaching 500 ± 0.02 at 7 camera frames. Furthermore wider convergence angles allowed for precise reconstruction of the camera extrinsic parameters. The improved camera projection matrices reduce triangulation error again resulting in improved accuracy. Once these thresholds are met, the SfM processes proved rather invariant to convergence angle and number of camera frames. This robustness permits greater operational freedom when collecting the target imagery. Despite the simulation's ability to provide detailed information regarding the operation of the SfM and MVS algorithms, the simulation was limited by both lack of occlusions and perfect knowledge of all correspondences. Therefore the second phase introduced real world synthetic scenes and images acquired from aerial surveillance platforms.

The 3-D rendering environment, Blender, was used to generate 2500 images encompassing 44 unique aerial flight profiles in which multiple combinations of the profiles resulted in the evaluation of 94 possible flight geometries. The flight profiles were reminiscent of typical reconnaissance patterns including linear, circular, and s-curve flight geometries including both static and agile sensors. The reconstructions generated from each pass were analyzed to quantify the likeness to the original scene including quantifiable measures such as root mean square accuracy and localized density of vertices within the target area. Linear profiles mounted with static cameras struggled to provide recognizable reconstructions of the target area at nadir or near nadir whereas dynamically controlled sensors excelled at these areas. Conversely, at oblique look angles, in which the look angle at the point of closest approach exceeds 45°, static cameras excelled whereas dynamic cameras failed. Circular orbits with the capability of imaging the target area from all sides only provided marginally better results despite the full target visibility and doubling of input images. S-curve profiles provided the richest set of images providing variability in both range to target, dynamically controlled sensors, and near 360° target visibility. The diversity of data allowed for extremely dense and accurate reconstructions only rivaled by the combination of two orthogonal linear static passes. The results of multiple passes permitted sufficient information to improve upon the independent linear passes. Furthermore, when only overlapping images from the two passes were included, results were further improved due to the elimination of anomalous 3-D points generated from a single pass. Finally, an investigation into the distribution of images within the algorithms permitted observation of a multi-modal distribution of image use within the reconstruction. This infers the algorithm heavily depends on only a select number of images at precise positions within the flight profiles. This dependency on critical image location was mapped for all linear profiles where it was shown the angle between the prominent images is inversely proportional to the look angle.

In summary, the interdependent nature between RMSDE accuracy, density of points, determination of camera motion, self calibration of camera focal length, and finally richness of image collection demand a rigorous study into the underpinnings of Structure from Motion. Multiple flight profiles demonstrated the ability to reconstruct ground targets with a high degree of accuracy and completeness. Linear flight paths with agile sensors proved the most feasible in terms of collection complexity and reconstruction accuracy suggesting application of these methods to overhead electro-optical imagery collected over denied areas. Rich image collections including both spatial and angular diversity in image correspondences to allow for the computation of a robust fundamental matrix and therefore an extremely accurate reconstruction. However, as was seen in the simulations, the process is robust to variances in the number of camera frames and convergence angles once the minimum thresholds are met. With these principles in mind, Structure from Motion derived 3-D reconstructions are a powerful tool for academic researchers, military operators, national decision makers, and the common individual with an interest in the 3-D world.

Appendix A. Levenberg Marquardt Algorithm

The Levenberg-Marquardt (LM) algorithm is an iterative technique which solves for the minimum of non-linear least squares problems. The algorithm leverages the strengths of both the steepest descent and Gauss-Newton regression methods and is nearly guaranteed to converge quicker than either function acting independently. For these reasons the Levenberg-Marquart technique has become the industry standard in the computer vision fields to rapidly converge to optimum solutions when adjusting camera parameters and vertex locations.

The steepest descent method incrementally approaches the minima by proportionally stepping along local negative gradient. In regions with significant gradients the steepest descent quickly converges; however, it slows in regions of low gradient. Functionally the method can be seen as

$$p_{i+1} = p_i - \nabla f\left(p_i\right) \tag{52}$$

where $\nabla f\left(p_i\right)$ is the gradient of the function at the current position [14]. It is preferable for a function to quickly step while in small gradient regions to facilitate rapid convergence and slowly increment towards the minimum when the gradient is steep to avoid overshooting the minimia [24].

The Gauss-Newton uses both the first and second derivatives to determine both the direction and magnitude of the local curvature. Expanding the local gradient through with a Taylor series expansions, it can be shown,

$$p_{i+1} = p_i - \left(\nabla^2 f\left(p_i\right)\right)^{-1} \nabla f\left(p_i\right) \tag{53}$$

where higher order terms are ignored [16].

Levenberg blended the two methods to capture the strengths to each. In doing so he created the function,

$$p_{i+1} = p_i - (H + \lambda I)^{-1} \nabla f(p_i) \tag{54}$$

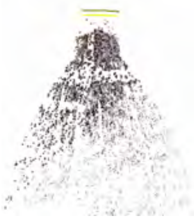
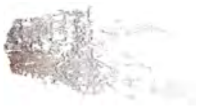
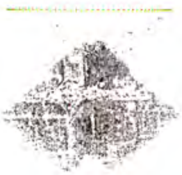
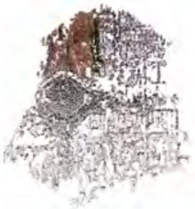
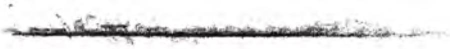
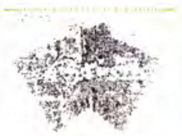
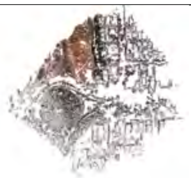
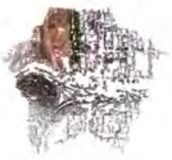
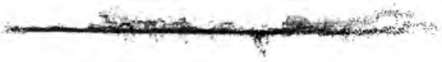
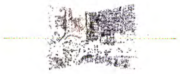
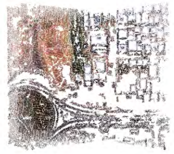
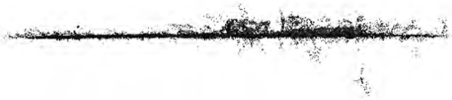
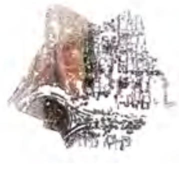
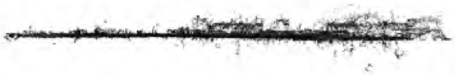
where the Hessian can be approximated by the Jacobian matrix, $H = \nabla^2 f(x) \approx J(x)^T J(x)$, and λ is the weighting factor emphasizing either the steepest descent or Gauss-Newton methods. During each iteration if the error is reduced and iteration accepted, λ is reduced, thereby deemphasizing the method of steepest descent. On the other hand, if the iteration is not accepted corresponding to an error increase, λ is increased.

Finally Marquardt improved upon Levenberg's refinements by using the Hessian matrix, H consisting of the second-order partial derivatives of the function describing the local function curvature, to include larger movements along the gradient according to the localized curvature,

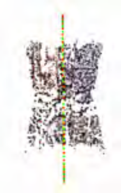
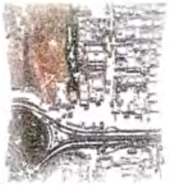
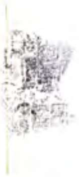
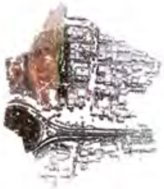
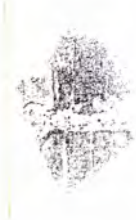
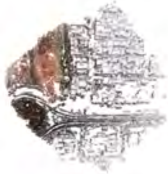
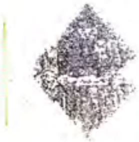
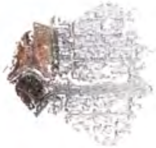
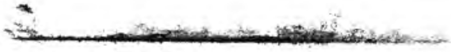
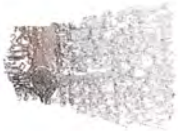
$$p_{i+1} = p_i - (H + \lambda diag[H])^{-1} \nabla f(p_i) \tag{55}$$

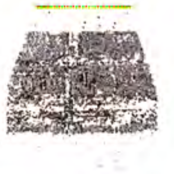
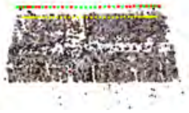
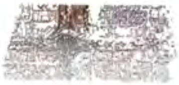
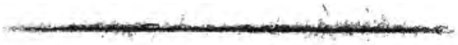
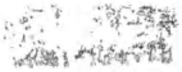
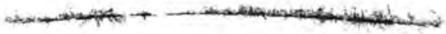
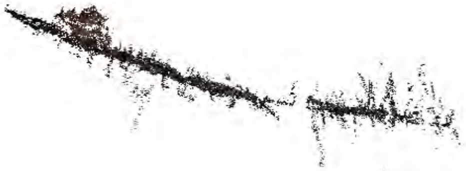
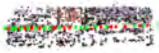
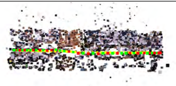
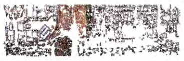
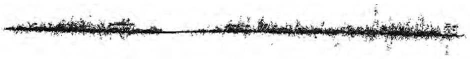
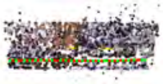
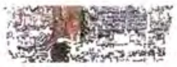
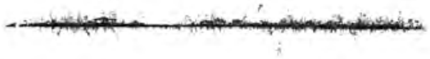
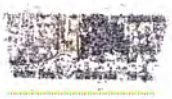
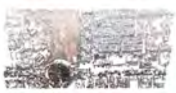
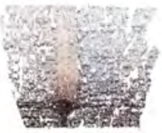
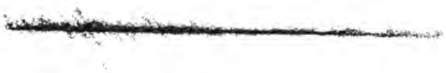
where the identity matrix, I, has been replaced with $diag[H]$ to appropriately scale the weighting factor, λ [15].

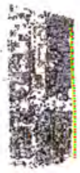
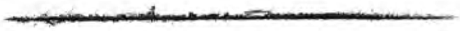
Appendix B. Results Summary

`linear_dynamic_xxx1000`			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
$60D$			
$45D$			
$30D$			
$15D$			
$0D$			
$-15D$			

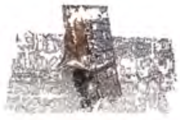
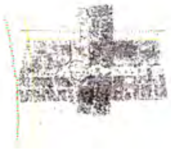
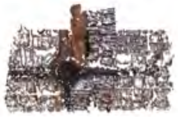
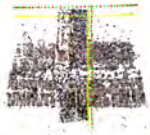
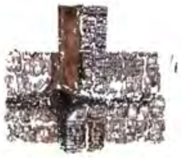
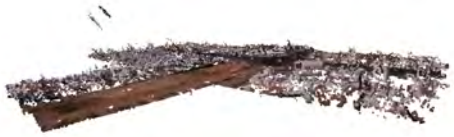
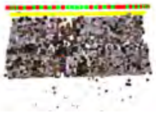
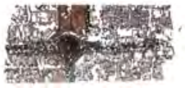
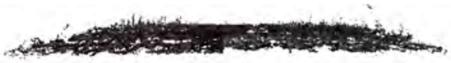
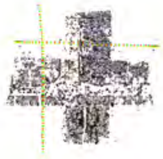
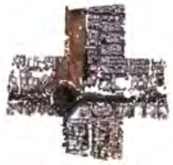
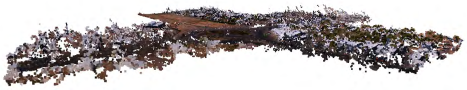
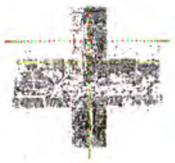
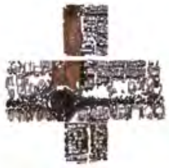
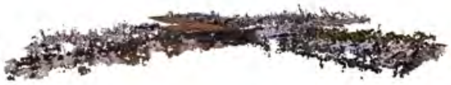
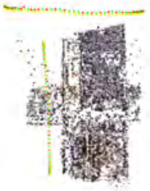
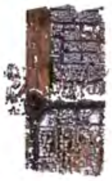
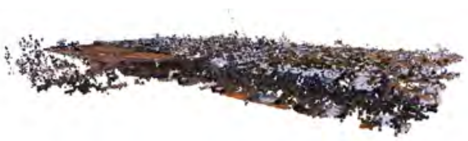
linear_dynamic_xxx1000			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
$-30D$			
$-45D$			
$-60D$			
$60H$			
$45H$			
$30H$			
$15H$			

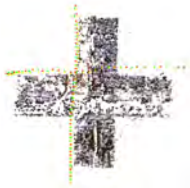
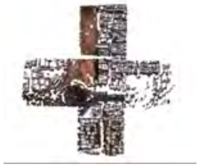
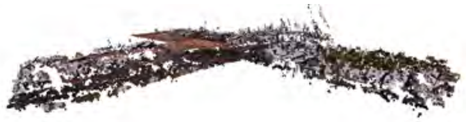
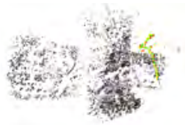
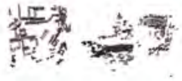
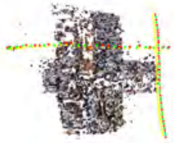
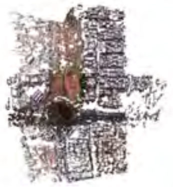
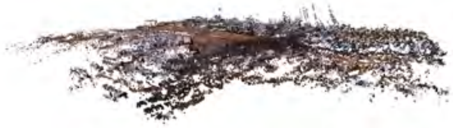
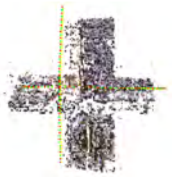
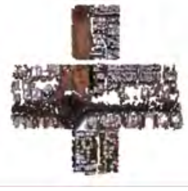
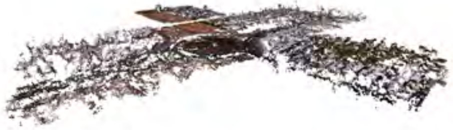
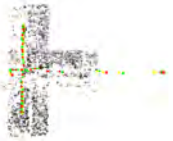
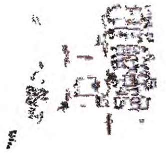
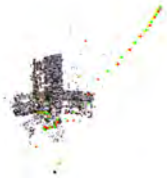
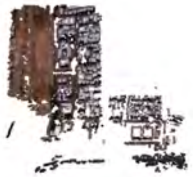
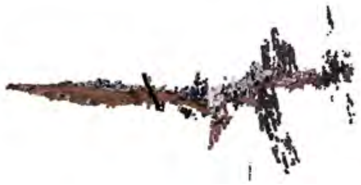
`linear_dynamic_xxx1000`			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
$0H$			
$-15H$			
$-30H$			
$-45H$			
$-60H$			

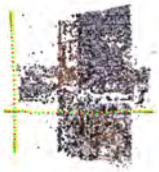
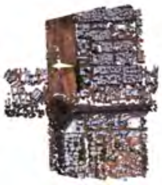
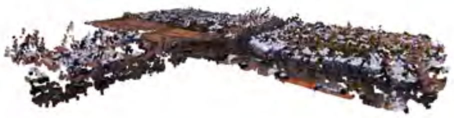
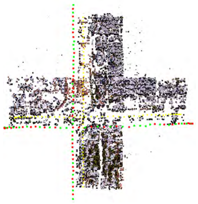
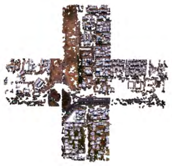
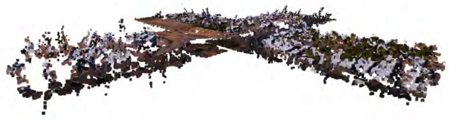
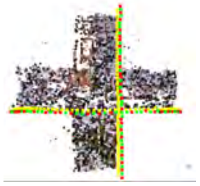
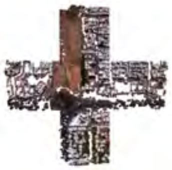
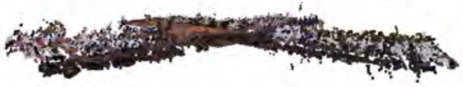
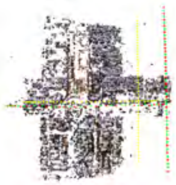
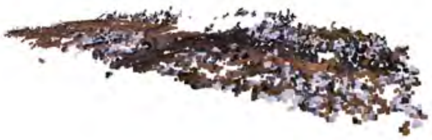
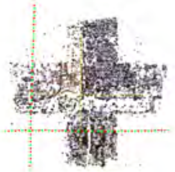
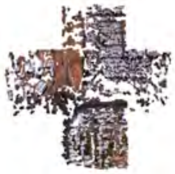
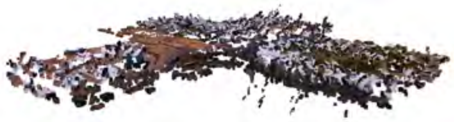
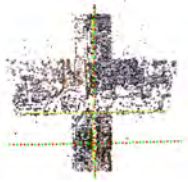
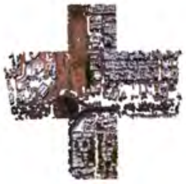
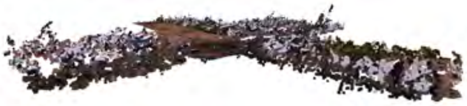
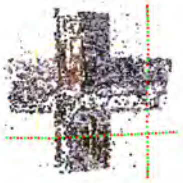
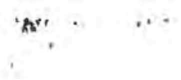
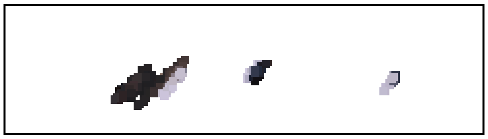
linear_static_xxx1000			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
$60D$			
$45D$			
$30D$			
$15D$			
$0D$			
$-15D$			
$-30D$			
$-45D$			
$-60D$			

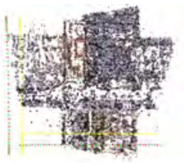
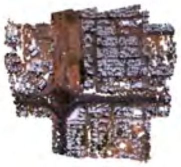
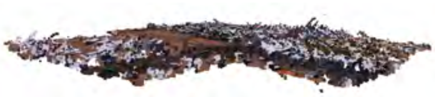
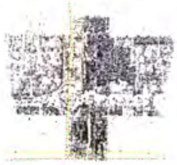
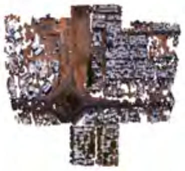
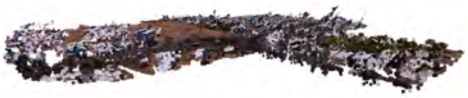
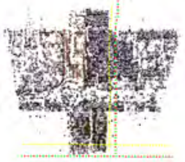
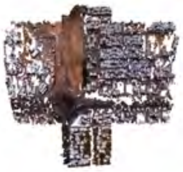
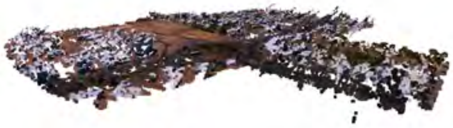
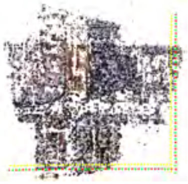
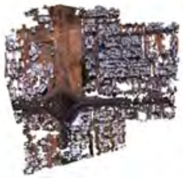
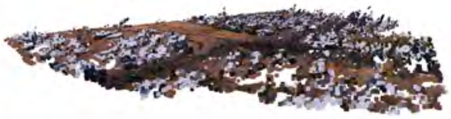
linear_static_xxx1000			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
$60H$			
$45H$			
$30H$			
$15H$			
$0H$			
$-15H$			

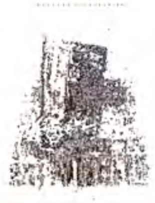
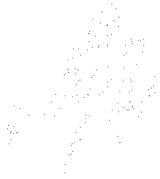
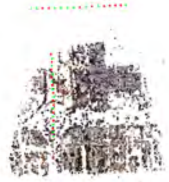
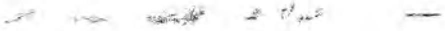
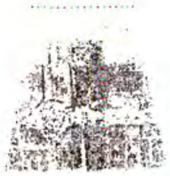
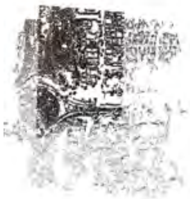
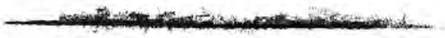
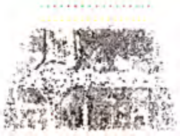
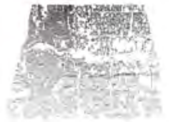
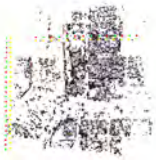
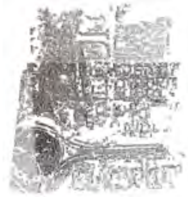
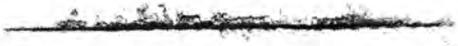
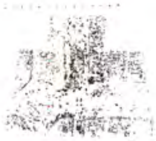
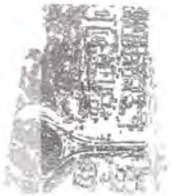
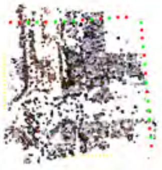
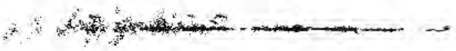
linear_static_xxx1000			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
$-30H$			
$-45H$			
$-60H$			

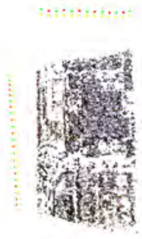
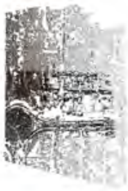
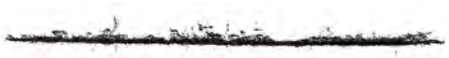
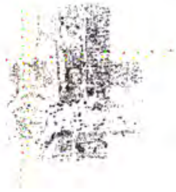
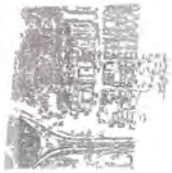
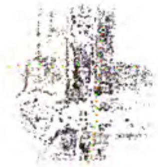
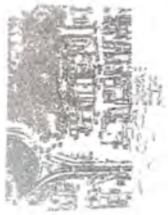
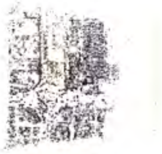
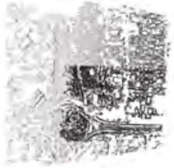
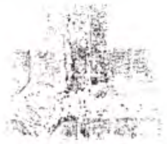
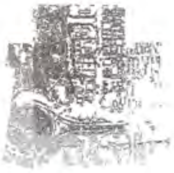
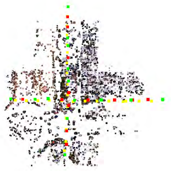
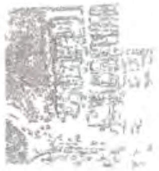
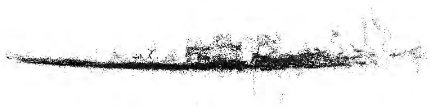
`linear_staticcross_xxDxxH`			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
$45D-45H$			
$45D-15H$			
$45D15H$			
$45D45H$			
$30D-30H$			
$30D0H$			
$30D30H$	*Failed*	*Failed*	*Failed*
$15D-45H$			

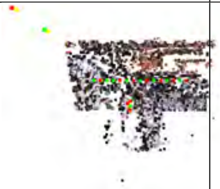
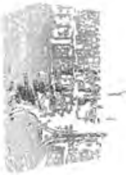
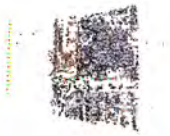
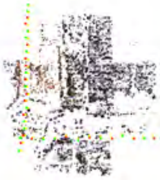
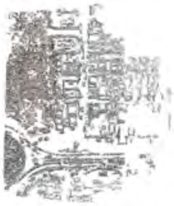
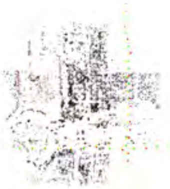
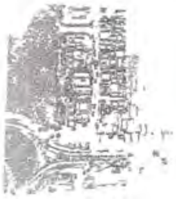
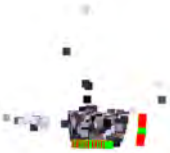
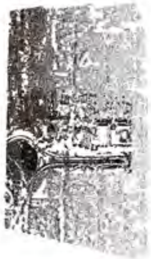
linear_staticcross_xxDxxH			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
$15D-15H$			
$15D15H$			
$15D45H$			
$0D-30H$			
$0D0H$			
$0D30H$			

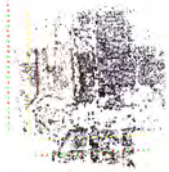
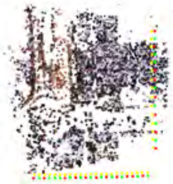
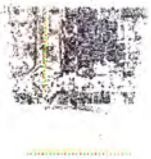
linear_staticcross_xxDxxH			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
$-15D-45H$			
$-15D-15H$			
$-15D15H$			
$-15D45H$			
$-30D-30H$			
$-30D0H$			
$-30D30H$			

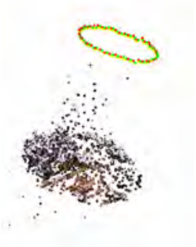
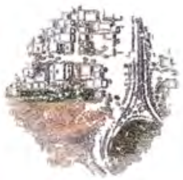
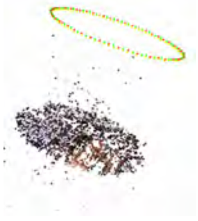
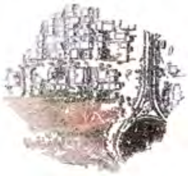
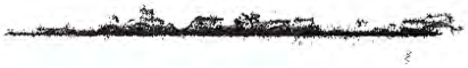
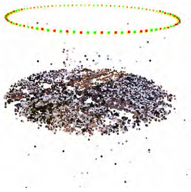
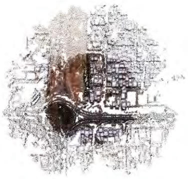
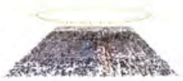
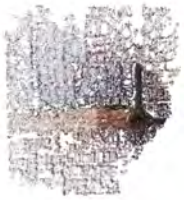
linear_staticcross_xxDxxH			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
$-45D-45H$			
$-45D-15H$			
$-45D15H$			
$-45D45H$			

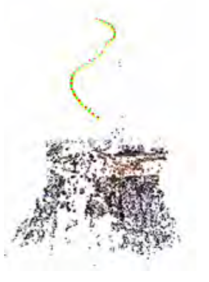
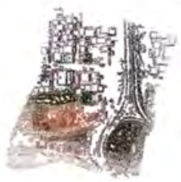
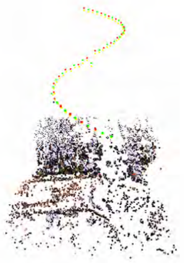
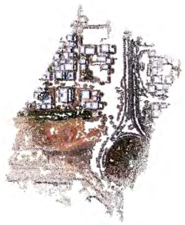
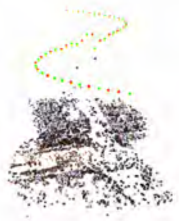
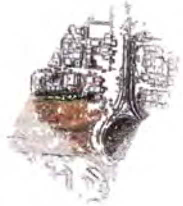
linear_staticcrosslimited_xxDxxH			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
$45D-45H$			
$45D-15H$			
$45D15H$			
$45D45H$			
$30D-30H$			
$30D0H$			
$30D30H$			

linear_staticcrosslimited_xxDxxH			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
$15D-45H$			
$15D-15H$			
$15D15H$			
$15D45H$			
$0D-30H$			
$0D0H$			

linear_staticcrosslimited_xxDxxH			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
$0D30H$			
$-15D-45H$			
$-15D-15H$			
$-15D15H$			
$-15D45H$			

linear_staticcrosslimited_xxDxxH			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
$-30D-30H$			
$-30D0H$			
$-30D30H$			
$-45D-45H$			
$-45D-15H$			
$-45D15H$			
$-45D45H$			

circular_dynamic_xxD1000			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
$15D$			
$30D$			
$45D$			
$60D$			

scurve_dynamic_xxD1000			
	Sparse Point Cloud	Dense Point Cloud	Dense Point Cloud-Ground Plane
15D			
30D			
45D			
60D		*Failed*	*Failed*

Appendix C. Analysis Summary

Table 11. Summary results of all aerial profiles: circular, s-curve, linear static, linear dynamic, linear static cross, and limited linear static cross. The number in parenthesis represents the overall ranking of the pass for the specific analytical technique, accuracy or completeness. The final column is the averaged ranking for a final scoring of the profile where a lower number represents a better score.

Aerial Profile	Accuracy	Completeness	Pass Score
circular_dynamic_15D1000	0.0432(16)	9265(12)	14.0(10)
circular_dynamic_30D1000	0.0707(43)	13654(2)	22.5(20)
circular_dynamic_45D1000	0.0641(37)	7685(29)	33.0(31)
circular_dynamic_60D1000	0.1701(74)	3214(65)	69.5(73)
scurve_dynamic_15D1000	0.0501(22)	9928(10)	16.0(15)
scurve_dynamic_30D1000	0.0231(1)	9014(15)	8.0(3)
scurve_dynamic_45D1000	0.0432(17)	10133(8)	12.5(7)
scurve_dynamic_60D1000	*fail*	*fail*	—
linear_dynamic_-60D1000	0.0983(60)	1102(79)	69.5(74)
linear_dynamic_-45D1000	0.0943(58)	2798(67)	62.5(63)
linear_dynamic_-30D1000	0.1014(61)	5208(53)	57.0(58)
linear_dynamic_-15D1000	0.0534(26)	7111(36)	31.0(28)
linear_dynamic_0D1000	0.0314(2)	8084(26)	14.0(12)
linear_dynamic_15D1000	0.0431(15)	8104(25)	20.0(18)
linear_dynamic_30D1000	0.0785(48)	6122(48)	48.0(49)
linear_dynamic_45D1000	0.1132(65)	3563(61)	63.0(64)
linear_dynamic_60D1000	*fail*	913(81)	—
linear_dynamic_-60H1000	0.0.0838(53)	1530(75)	64.0(65)
linear_dynamic_-45H1000	0.0914(55)	4147(55)	55.0(57)
linear_dynamic_-30H1000	0.0550(28)	6230(47)	37.5(38)
linear_dynamic_-15H1000	0.0603(33)	9307(11)	22.0(19)
linear_dynamic_0H1000	0.0354(7)	9072(14)	10.5(4)
linear_dynamic_15H1000	0.0449(18)	8028(27)	22.5(21)
linear_dynamic_30H1000	0.0668(39)	6303(44)	41.5(43)
linear_dynamic_45H1000	0.0708(44)	3859(59)	51.5(56)
linear_dynamic_60H1000	0.0930(57)	1412(77)	67.0(71)
linear_static_-60D1000	0.0833(52)	1258(78)	65.0(69)
linear_static_-45D1000	0.0.0546(27)	2769(68)	47.5(48)
linear_static_-30D1000	0.0566(30)	3767(60)	45.0(46)
linear_static_-15D1000	0.0670(40)	3965(58)	49.0(52)
linear_static_0D1000	*fail*	711(83)	—
linear_static_15D1000	*fail*	629(84)	—
linear_static_30D1000	*fail*	1082(80)	—
linear_static_45D1000	0.1654(72)	2098(72)	72.0(76)
linear_static_60D1000	0.0920(56)	1777(73)	64.5(68)

Aerial Profile	Accuracy	Completeness	Pass Score
linear_static_-60H1000	0.0872(54)	1615(74)	64.0(66)
linear_static_-45H1000	0.1213(67)	3460(62)	64.5(67)
linear_static_-30H1000	0.0522(25)	6690(42)	33.5(33)
linear_static_-15H1000	0.1026(62)	8251(22)	42.0(45)
linear_static_0H1000	*fail*	717(82)	—
linear_static_15H1000	0.0590(32)	6895(38)	35.0(34)
linear_static_30H1000	*fail*	3433(63)	—
linear_static_45H1000	0.0797(49)	3158(66)	57.5(59)
linear_static_60H1000	0.1760(75)	1519(76)	75.5(78)
linear_staticcross_45D-45H	*fail*	*fail*	—
linear_staticcross_45D-15H	0.0675(42)	10194(7)	24.5(22)
linear_staticcross_45D15H	0.1099(64)	5904(52)	58.0(60)
linear_staticcross_45D45H	0.0973(59)	2273(71)	65.0(70)
linear_staticcross_30D-30H	0.0672(41)	7273(34)	37.5(39)
linear_staticcross_30D0H	0.0356(8)	6353(43)	25.5(24)
linear_staticcross_30D30H	*fail*	*fail*	—
linear_staticcross_15D-45H	0.1323(69)	7509(31)	50.0(54)
linear_staticcross_15D-15H	0.1051(63)	8916(16)	39.5(42)
linear_staticcross_15D15H	*fail*	*fail*	—
linear_staticcross_15D45H	0.1165(66)	10374(6)	36.0(35)
linear_staticcross_0D-30H	0.0643(38)	7244(35)	36.5(36)
linear_staticcross_0D0H	*fail*	*fail*	—
linear_staticcross_0D30H	0.1530(71)	2743(69)	70.0(75)
linear_staticcross_-15D-45H	0.0503(23)	11586(4)	13.5(9)
linear_staticcross_-15D-15H	0.0340(5)	8491(19)	12.0(6)
linear_staticcross_-15D15H	0.1426(70)	11999(3)	36.5(37)
linear_staticcross_-15D45H	*fail*	*fail*	—
linear_staticcross_-30D-30H	0.1872(77)	2463(70)	73.5(77)
linear_staticcross_-30D0H	0.0493(20)	6245(45)	32.5(30)
linear_staticcross_-30D30H	*fail*	*fail*	—
linear_staticcross_-45D-45H	0.0761(46)	5942(51)	48.5(50)
linear_staticcross_-45D-15H	0.0427(14)	15261(1)	7.5(2)
linear_staticcross_-45D15H	0.0494(21)	9993(9)	15.0(14)
linear_staticcross_-45D45H	0.0765(47)	5176(54)	50.5(55)
linear_staticcross_limited_45D-45H	*fail*	*fail*	—
linear_staticcross_limited_45D-15H	*fail*	*fail*	—
linear_staticcross_limited_45D15H	0.0616(34)	7794(28)	31.0(27)
linear_staticcross_limited_45D45H	0.1678(73)	3295(64)	68.5(72)
linear_staticcross_limited_30D-30H	0.1289(68)	7559(30)	49.0(51)
linear_staticcross_limited_30D0H	0.0833(51)	7349(33)	42.0(44)
linear_staticcross_limited_30D30H	*fail*	*fail*	—
linear_staticcross_limited_15D-45H	0.1833(76)	6235(46)	61.0(62)

Aerial Profile	Accuracy	Completeness	Pass Score
linear_staticcross_limited_15D-15H	0.0759(45)	8605(17)	31.0(26)
linear_staticcross_limited_15D15H	0.0322(3)	8225(23)	13.0(8)
linear_staticcross_limited_15D45H	0.0504(24)	7067(37)	30.5(25)
linear_staticcross_limited_0D-30H	0.0364(9)	9079(13)	11.0(5)
linear_staticcross_limited_0D0H	0.0639(36)	6731(41)	38.5(40)
linear_staticcross_limited_0D30H	0.0552(29)	6092(49)	39.0(41)
linear_staticcross_limited_-15D-45H	0.2254(78)	6785(40)	59.0(61)
linear_staticcross_limited_-15D-15H	0.0325(4)	8206(24)	14.0(11)
linear_staticcross_limited_-15D15H	0.0424(13)	8426(20)	16.5(16)
linear_staticcross_limited_-15D45H	0.0504(24)	7067(37)	30.5(25)
linear_staticcross_limited_-30D-30H	0.0407(11)	6868(39)	25.0(23)
linear_staticcross_limited_-30D0H	0.0458(19)	8274(21)	20.0(17)
linear_staticcross_limited_-30D30H	0.0577(31)	7455(32)	31.5(29)
linear_staticcross_limited_-45D-45H	0.0623(35)	3994(57)	46.0(47)
linear_staticcross_limited_-45D-15H	0.0353(6)	11495(5)	5.5(1)
linear_staticcross_limited_-45D15H	0.0421(12)	8598(18)	15.0(13)
linear_staticcross_limited_-45D45H	0.0623(35)	3994(57)	46.0(47)

Appendix D. Software Guide

D.1 Bundler

This section details the installation procedures used by the author to install and run Bundler to generate the necessary sparse point clouds and camera parameters.

D.1.1 Bundler Installation.

These steps are reiterated from the Bundler homepage [2] with several modifications specific to location of necessary libraries and support executables.

1. Download the necessary binary distribution package, `bundler-v0.3-binary.zip` from *http://phototour.cs.washington.edu/bundler/* and extract it into a directory, thereafter referred to as the `BASE_PATH`.

2. Bundler relies on bash and perl installations. The easiest environment to run these scripts is through cygwin. When installing cygwin, all packages must be installed which is not the default option.

3. Download `sift.exe` from *http://www.cs.ubs.ca/∼ lowe/keypoints/* and copy file into the `BASE_PATH\bin` folder.

4. `RunBundler.sh` must be modified to include the present working directory.

   (a) Change line 17 from `$BASE_PATH=$(dirname$(which $0))` to `BASE_PATH=$PWD`

D.1.2 Bundler Operation.

1. Prior to running Bundler all images must be placed in a common folder and resized to ensure dimensions do not exceed 2000-by-2000. This requirement is specifically for the `sift.exe` operation.

2. Open cygwin and navigate to folder containing `RunBundler.sh`.

3. Enter `sh RunBundler.sh filepath/` into the command line.

4. Bundler will immediately begin and follows four steps:

   (a) Extract focal length from image metadata.

(b) Extracts keypoints from each image.

(c) Matches keypoints from image to image. This steps requires additional time for each picture in the data set since the image is being matched to all previous images.

(d) Running Bundler. At this point Bundler is determining camera pose and triangulating all 3-D vertices.

Upon conclusion a new folder will be created in the directory containing `runbundler.sh`. The contents of the folder will contain a `bundle.out` file containing all the information from the sparse point cloud generation including camera parameters and 3-D vertices. Additionally a series of `bundle_xxx.out` and `points_xxx.out` files. Each file contains additionally information as they are processed and outputted from bundler as the program in running. Select the `points_xxx.out` with the largest number, typically the number of images used, and open with Meshlab or CloudCompare to view results [8, 4].

D.2 PMVS2/CMVS

This effort used a version of PMVS2/CMVS specifically modified for the Microsoft Window's operating systems developed by Pierre Moulon.

D.2.1 PMVS2/CMVS Installation.

1. Download PMVS2/CMVS binary packages from *http://blog.neonascent.net/archives/bundler-photogrammetry-package/*. Specifically the `SFM.zip` toolkit was used.

2. Extract all files.

D.2.2 PMVS2/CMVS Operation.

1. Create a folder in which raw `.jpeg` images and bundler output files co-reside.

2. Copy the following PMVS/CMVS files into the folder.
   (a) `denseRecon.vbs`
   (b) `denseRecon_batch.vbs`
   (c) `EXIFwrite.vbs`
   (d) `makelist.bat`
   (e) `matrix.bat`
   (f) `matrix-ListWSize.bat`
3. Double click `makelist.bat` to generate a `list.txt` which contains the filenames of all images within the folder.
4. Click `denseRecon.vbs` and follow instructions requesting variable inputs. See Methodology chapter for additional guidance.
5. Click Run

PMVS2/CMVS will execute and upon completion will generate a host of new files. The most important files pertaining to the dense point cloud are located in the `...\PMVS\models` folder. Each `option-000x.txt.patch` and `option-000x.txt.ply` contain the necessary information to view and analyze the dense point cloud. Within the `option-000x.txt.patch` patch information including the location, normal projection, and camera frames the patch was extracted from. The `option-000x.txt.ply` simply provides a direct viewing capability of the point cloud via Meshlab or CloudCompare. Furthermore when multiple `option-000x.txt.patch` or `option-000x.txt.ply` files exist they must be merged to be view the point cloud in its entirety.

D.3 Blender Image Acquisition

Blender, a 3-D animation studio, offers the ability to generate images from a variety of positions of user generated 3-D scenes or objects.

D.3.1 Blender Installation.

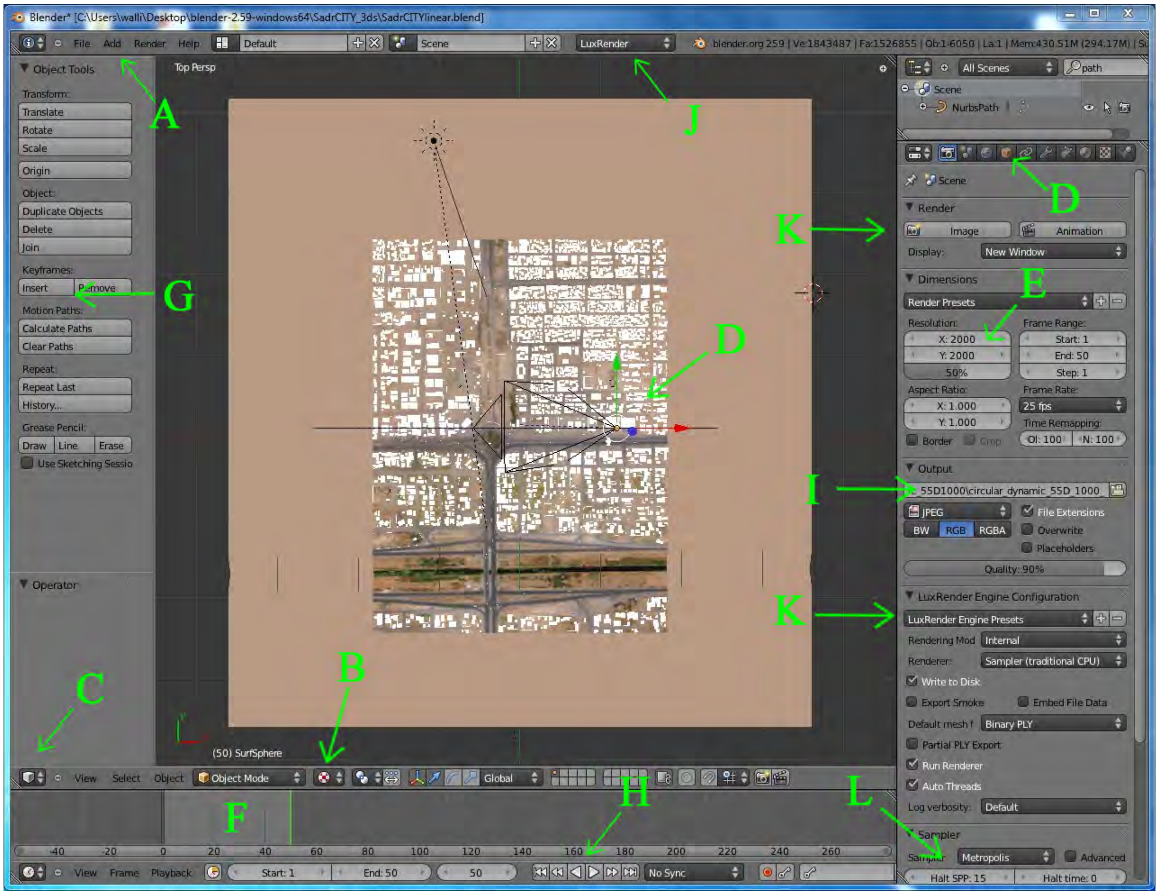
Blender and LuxRender installation is relatively straight forward as both software products have reached a satisfactory level of maturity.

1. Download Blender 2.5 from *http://www.blender.org/download/get-blender/*. Note at the time of this installation Blender 2.51 was the current distribution.
    (a) Install per Blender instructions.
2. Download LuxRender from *http://www.luxrender.net/en_GB/blender_2_5*.
    (a) Install per LuxRender instructions.
    (b) After installation LuxBlend25 must be activated with Blender.
        i. Open Blender and click `User Preferences`→`Add-Ons`→`Render`.
        ii. Click LuxRender to activate.
        iii. Proceed to the main Blender page and select LuxRender from the drop down rendering engine selection list, see label K in Figure 68.

D.3.2 Blender Operation.

The following shortcuts will greatly ease familiarization process with Blender.

- Right click selects an object in 3D space.
- Click and hold middle mouse button to rotate object, holding `Shift` at the same time will translate the object without rotating.
- Numberpad 7 will automatically center the map in to look straight down
- Numberpad (1-6,8,9) will automatically reorient the map to a variety of perspectives
- Numberpad 0 will show camera view.
- It is suggested to move objects by left clicking and holding the axes arrows once the object is selected, thereby restricting movement to the specific axis.



Render: Image, Animation
Display: New Window
Dimensions
Render Presets
Resolution: X: 2000, Y: 2000, 50%
Frame Range: Start: 1, End: 50, Step: 1
Aspect Ratio: X: 1.000, Y: 1.000
Frame Rate: 25 fps
Time Remapping: Ol: 100, N: 100
□ Border □ Crop
Output
_55D1000\circular_dynamic_55D_1000_
JPEG
BW RGB RGBA
☑ File Extensions
□ Overwrite
□ Placeholders
Quality: 90%
LuxRender Engine Configuration
LuxRender Engine Presets
Rendering Mod: Internal
Renderer: Sampler (traditional CPU)
☑ Write to Disk
□ Export Smoke □ Embed File Data
Default mesh f: Binary PLY
□ Partial PLY Export
☑ Run Renderer
☑ Auto Threads
Log verbosity: Default
Sampler
Sampler: Metropolis □ Advanced
Halt SPP: 15 Halt time: 0
Start: 1 End: 50 50

Figure 68. Command window for blender.

1. Open `blender.exe`.
2. Click `File`→`Open`→`SadrCITY.blend` or import necessary `.obj` file, see label A.
3. Ensure ground, vehicle, and landscaping textures have been applied by selecting `Viewport Shading`→`Textured`, see label B. If background appears fuschia in color follow the sub-steps. Note Textured shading requires significant memory. If computer lags increases revert back to Solid texture. If ground texture has not been preloaded follow these steps:
   (a) Click `Editor`→`Text Editor`, see label C.
   (b) Open existing text block `Alt-O`.
   (c) Navigate to and double click `Add_Texture_to_Sadr_City_Model.py`.
   (d) Ensure path variable contains the file location for the `.jpg` texture files.

    (e) Save script, `Alt-S`.

    (f) Run script, `Alt-P`.

    (g) Verify successful application of textures by returning to 3D View and Textured Shading.

4. We first designate the number of images we wish to acquire.

    (a) To do so select the camera icon along the properties bar (see label D), navigate to the dimension tab, and ensure your frame range starts with 1, terminates at the desired end frame, and step size is 1, see label E.

    (b) Go to the timeline and ensure frame 1 is selected, see label F.

5. Add a light source to the file by clicking `Add`→`Lamp`→`Sun`. Position is irrelevant as a sun simulation lamp is place at a point of infinity to simulate parallel rays of radiance.

    (a) Select the sun in the 3D view represented by a small dot with eight rays radially emanating.

    (b) Select Object icon from the property bar (see label D) and set rotation angle for the incident sunlight. All images rendered in this effort used a sun angle of $-45°$ in the x-direction.

6. Add a path for our camera to follow by clicking `Add`→`Curve`→`Path`.

    (a) With path still selected you can position the path at any point in the working environment. For a direct nadir pass over the target of `HB11_House1` input a location of $(-0.02798, 0.140, 1.000)$ in the object tab of the properties toolbar. An altitude of 1 was maintained throughout this effort.

7. Add a camera to follow the path by clicking `Add`→`Camera`.

    (a) First click movie camera icon in the property bar, see label D, and ensure the lens is in perspective not orthographic mode and select appropriate focal length ($50mm$ used).

    (b) With camera selected press `Shift` and right click the path. Path will now be highlighted in orange whereas the camera will assume a redish-orange coloration.

    (c) Press `Ctrl-P` to set cameras set parent `Follow Path`.

   (d) If successful a dotted line will now connect the frustrum of the camera to the beginning of the path.

8. Now we must properly position the camera on the path and insert keyframes.

   (a) Select the camera by right clicking on any of the lines depicting the camera frustum.

   (b) By left clicking and holding on each of the colored coordinate axes drag the camera to the location indicated by the dotted line. It is easiest to resolve the cameras z position by inputting 1 within the object tab of the property toolbar. For this effort I simply indicted a x-location of -1.

   (c) Click `Insert Keyframe`→`Location`, see label G. Location box will appear yellow and green line on timeline will also turn yellow.

   (d) Return to timeline and select the last frame, see label F.

   (e) Camera will automatically translate to the mirror image on the other side of the target area.

   (f) Click `Insert Keyframe`→`Location`

9. If a dynamic camera is required.

   (a) Right click camera frustum to select.

   (b) On the property bar, select the `chain link icon`→`Add Constraint`→`Track To`. For the target select HB11_House1 and to correctly orient camera input To: -Z and Up: Y

10. Ensure camera moves steps linearly throughout path.

    (a) Select Graph Editor from the editor button and ensure the red line is linear. This line slope determines the stepping distance between each camera frame. The default line in not linear as to facilitate a smooth start and finish for the camera motion. This adds unnecessary complications to extracting the real camera location.

    (b) Return to 3D view.

11. At this point the path should be complete. Test by clicking play arrow, see label H, on the timeline and ensure the camera iteratively steps from the starting point to the end point. Additionally pressing Numberpad 0 will show view from camera focal point.

12. Setup of the image acquisition sequence.

    (a) Click camera icon from property bar.

    (b) Select required resolution in the dimensions box, do not dismiss the scaling percentage. Set Scaling to 100% for full resolution as indicated above.

    (c) Select save location and name in the output tab, see label I. Blender will automatically add a camera number to each individual frame.

    (d) Configuring LuxRender Engine, see label J.

        i. Ensure LuxRender is selected rendering engine, see label K.

        ii. Select Metropolis Light Transport (unbiased-recommended)

        iii. Rendering Mode: Internal (Allows for animation collection without interruption)

        iv. Renderer: Sampler (traditional CPU) To date the hybrid or GPU render assist does not work

        v. Within the sampler tab select Halt SPP: 15. This dictates how may samples per pixel are required. Larger numbers result in less noise but rendering time increases exponentially.

13. Running the animation.

    (a) Return to the camera icon in the property bar, see label L, and there are two options: image or animation.

        i. *Image* will only render a single photograph and is often a useful verification all parameters have been correctly implemented.

        ii. *Animation* processes all images sequentially

Blender will output several files including the `.jpeg` files used in the reconstruction. First a `.lxs` file contains the necessary information Blender passed to LuxRender to properly render the scene. LuxRender in return provides a `.png` back to Blender for final processing which results in the final `.jpeg` images.

Bibliography

[1] “Blender 2.5, 3D Computer Graphics Rendering Software”. ”Blender” Homepage. Available at
http : //www.blender.org.

[2] “Bundler: Structure from Motion(SfM) for Unordered Image Collections”. Bundler Homepage. Available at
http : //phototour.cs.washington.edu/bundler/.

[3] “Camera Calibration Toolbox for MATLAB”. Camera Calibration Homepage. Available at
http : //www.vision.caltech.edu/bouguetj/calib_doc/.

[4] “CloudCompare, 3D point cloudl and mesh processing software”. ”CloudCompare” Homepage. Available at
http : //www.danielgm.net/cc/.

[5] “Clustering Views for Multi-View Stereo (CMVS)”. CMVS Homepage. Available at
http : //grail.cs.washington.edu/software/cmvs/.

[6] “Google Earth”. Google Earth Homepage. Available at
http : //earth.google.com.

[7] “LuxRender , GPL Physically Based Renderer”. ”LuxRender” Homepage. Available at
http : //www.luxrender.net/en_GB/blender_2_5.

[8] “Meshlab”. ”Meshlab” Homepage. Available at
http : //meshlab.sourceforge.net/.

[9] “Microsoft Photosynth”. Microsoft Photosynth Homepage. Available at
http : //photosynth.net.

[10] “Ohio Geographically Reference Information Program”. OSIP Imagery and Elevation Data Download Website. Available at
http : //gis4.oit.ohio.gov/osiptiledownloads/default.aspx.

[11] “Patch-based Multi-view Stereo Software (PMVS - Version 2)”. PMVS Homepage. Available at
http : //grail.cs.washington.edu/software/pmvs/.

[12] “Sample Code”. ”An Invitation to 3D Vision” Homepage. Available at
http : //cs.gmu.edu/ ∼ kosecka/bookcode.html.

[13] Agarwal, Sameer, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven Seitz, and Richard Szeliski. "Building Rome in a Day". *Communications of the ACM*, 54(10):105–112, 2011.

[14] Avriel, Mordecai. *Nonlinear Programming: Analysis and Methods*. Prentice Hall, Englewood Cliffs, NJ, 1976.

[15] Bazaraa, Mokhtar S., Hanif D. Sherali, and C.M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley and Sons, Inc., New York, 1979.

[16] Bertsekas, Dimitri P. *Nonlinear Programming*. Athena Scientific, Belmont, Mass, 1999.

[17] Fischler, Martin and Robert Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis adn Automated Cartography". *Communications to the ACM*, 24(6):381–395, 1981.

[18] Furukawa, Yasutaka, Brian Curless, Steven Seitz, and Richard Szeliski. "Towards Internet-scale Multi-view Stereo". *Computer Vision and Pattern Recognition (CVPR)*, 2010 IEEE Conference:1434–1441, 2010.

[19] Furukawa, Yasutaka and Jean Ponce. "Accurate, Dense, and Robust Multiview Stereopsis". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

[20] Goesele, Michael, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M Seitz. "Mulit-View Stereo for Community Photo Collections". *Proceedings from IEEE International Conference on Computer Vision*. October 2007.

[21] Graham, Paul R. *Determination of Structure from Motion Using Aerial Imagery*. Master's thesis, Air Force Institute of Technology, 2005.

[22] Harris, Chris and Mike Stephens. "A Combined Corner and Edge Detector". *Proceedings of the 4th Alvery Vision Conference*.

[23] Hartley, Richard and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2010.

[24] Lach, Stephen R. *Semi-Automated DIRSIG Scene Modeling from 3D Lidar and Passive Imagery*. Ph.D. thesis, Rochester Institute of Technology, 2008.

[25] Lourakis, Manolis and Antonis Argyros. *The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm*. Technical Report TR-340, Institute of Computer Science, 2004.

[26] Lourakis, Manolis and Antonis Argyros. "SBA: A Software Package for Generic Sparse Bundle Adjustment". *ACM Transactions on Mathematical Software*, 36(1), 2009.

[27] Lowe, David. "Distinctive Image Features from Scale-Invariant Keypoints". *International Journal of Computer Vision.*

[28] Ma, Yi, Sefano Soatto, Jana Kosecka, and S. Shankar Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models.* Springer-Verlag, New York, New York, 2010.

[29] McNeil, Gomer. *Photographic Measurements.* Pitnam Publishing Corporation, New York, 1952.

[30] Moravec, H. "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover". *Tech Report CMU-RI-TR-3 Carnegie-Mellon University.*

[31] Mount, David. *ANN Programming Manual.* Technical report, University of Maryland, 2010.

[32] Palmer, Stephen. *Vision Science: Photons to Phenomenology.* MIT Press, Cambridge, Massachusetts, 1999.

[33] Peter Mountney, Danail Stoyanov and Guang-Zhong Yang. "Three-Dimensional Tissue Deformation Recovery and Tracking". *IEEE Signal Processing Magazine*, 24(4):14–24, 2010.

[34] Torr, Philip, Andrew Fitzgibbon, and Andrew Zisserman. "The Problem of Degeneracy in Structure and Motion Recovery from Uncalibrated Image Sequences". *International Journal of Computer Vision*, 32(1):27–44, 1999.

[35] Walli, Karl C. *Relating Multimodal Imagery Data in 3D.* Ph.D. thesis, Rochester Institute of Technology, 2010.

[36] Wolf, Paul R. and Bon A. Dewitt. *Elements of Photogrammetry with Applications in GIS.* McGraw Hill, Boston, Massachusetts, 2004.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)* 22–03–2012	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* Sep 2010 — Mar 2012

4. TITLE AND SUBTITLE

3-D Scene Reconstruction from Aerial Imagery

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Ekholm, Jared M., Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/APPLYPHY/ENP/12-M03

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RYA (Dr. Michael Talbert)
2241 Avionics Circle
Wright-Patterson AFB, OH 45433
937-582-8506, michael.talbert@wpafb.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RYA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

3-D scene reconstructions derived from Structure from Motion (SfM) and Multi-View Stereo (MVS) techniques were analyzed to determine the optimal reconnaissance flight characteristics suitable for target reconstruction. In support of this goal, a preliminary study of a simple 3-D geometric object facilitated the analysis of convergence angles and number of camera frames within a controlled environment. Reconstruction accuracy measurements revealed at least 3 camera frames and a 6 convergence angle were required to achieve results reminiscent of the original structure. The central investigative effort sought the applicability of certain airborne reconnaissance flight profiles to reconstructing ground targets. The data sets included images collected within a synthetic 3-D urban environment along circular, linear, and s-curve aerial flight profiles equipped with agile and non-agile sensors. S-curve and dynamically controlled linear flight paths provided superior results, whereas with sufficient data conditioning and combination of orthogonal flight paths, all flight paths produced quality reconstructions under a wide variety of operational considerations.

15. SUBJECT TERMS

Structure from Motion, Multi-View Stereo, 3D Scene Reconstruction, Epipolar Geometry, Photogrammetry

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			LtCol Karl Walli
U	U	U	U	182	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255-3636, x4333; karl.walli@afit.edu